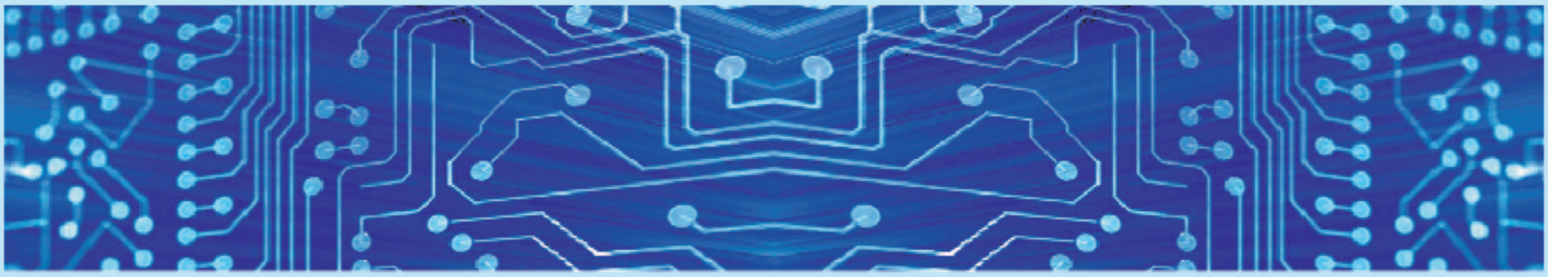
Jordanian Journal of Computers and Information Technology

December 2020 VOLUME 06 NUMBER 04
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

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

Editorial Board Support Team

JJCIT Address

Website: www.jjcit.org
Email: jjcit@psut.edu.jo
Address: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. Box: 1438 Amman 11941 Jordan
Telephone: +962-6-5359949
Fax: +962-6-7295534

"Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund".

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي".

HYBRID BLOCKCHAIN

Hazem W. Marar[1] and Rosana W. Marar[2]

(Received: 10-May-2020, Revised: 28-Jun.-2020 and 20-Jul.-2020, Accepted: 8-Aug.-2020)

ABSTRACT

Blockchain is a revolutionary technology that gained widespread popularity since the emergence of crypto-currencies. The potential uses of Blockchain surpassed digital currency into a wider space that includes the Internet of Things (IoT), security applications and smart embedded systems, among others. As the number of Blockchain users increases, several drawbacks start to emerge, since Blockchains consume excessive amounts of energy to store and manipulate data. Furthermore, the limited scalability nature of Blockchains due to their massive storage requirements might become an issue. To improve the overall performance, several challenges in the current Blockchain structure should be tackled. This paper presents a hybrid system architecture that combines the distributed nature of Blockchains with the centralized feature of servers. Users will connect to servers via personal Blockchains, while servers will share a chain of Blockchains to ensure integrity and security. This will significantly decrease the storage requirements of end-users and enhance the scalability of networks. Businesses will highly benefit from this proposed structure, since it creates a reliable scalable business model.

KEYWORDS

Blockchain, Bitcoin, Crypto-currency, Distributed, E-commerce, Hybrid.

1. INTRODUCTION

Since the emergence of Bitcoin in 2008 [1], crypto-currencies and hence Blockchain technology have gained widespread popularity. The technology features enable it to become the infrastructure for a new generation of internet interactions that include secure online payments [2]-[3], data exchange and transaction of digital assets [2]-[4]. Blockchain provides a decentralized, open, Byzantine fault-tolerant transaction mechanism [5]-[7]. Users can consider Blockchain as a sort of data structure that consists of an ever increasing number of blocks linked together through cryptography. Each block includes a cryptographic hash of the previous block, a timestamp and data that users wish to exchange throughout the network [8]-[9]. Data blocks are shared among users and not saved on a centralized server. The chain of blocks continuously grows when new blocks are appended into it and this change will be reflected to all users within the chain. Hence, by design, Blockchain is relatively resistant to data modification. However, in order to append a new block to the chain, computers, called miners, compete and run a complex hashing algorithm trying to produce a valid block hash. This will dissipate huge amounts of disproportionate power and time. Furthermore, the ever increasing chain of blocks will require massive storage capabilities from all users, limiting the scalability of such technology. Blockchains are governed by a consensus algorithm used as a mechanism to achieve the necessary agreement on the validity of data among distributed processes. With the recent advances in Blockchain technology, numerous consensus algorithms were proposed to make endpoints reach an agreement on the order and state of blocks of transactions and update the distributed ledger accordingly [10]-[11]. In this paper, a hybrid Blockchain architecture that is suitable for online banking and e-commerce platforms is proposed. The proposed solution addresses the storage and power consumption requirements while improving the Blockchain's integrity and security. The paper is organized as follows: Section 2 introduces the Blockchain technology and its methodology, whereas Section 3 illustrates the architecture of the proposed hybrid solution and Section 4 concludes the paper.

2. BLOCKCHAIN TECHNOLOGY

Blockchain is a distributed transactional database that is governed by network consensus and secured by advanced cryptography. As illustrated in Figure 1, a Blockchain consists of a series of datasets that are composed of chains of data blocks. Each block holds several transactions (TX_n). Once a set of

[1] Hazem W. Marar is with Department of Electrical Engineering, Princess Sumaya University for Technology, Amman, Jordan. Email: h.marar@psut.edu.jo

[2] Rosana W. Marar is with Department of Computer Graphics and Animation, Princess Sumaya University for Technology, Amman, Jordan. Email: r.marar@psut.edu.jo

transactions is complete, an additional block is appended to the chain, hence representing a complete ledger of transactions history. The chain of blocks continuously grows when new blocks are appended into it and this change will be reflected to all users within the chain.

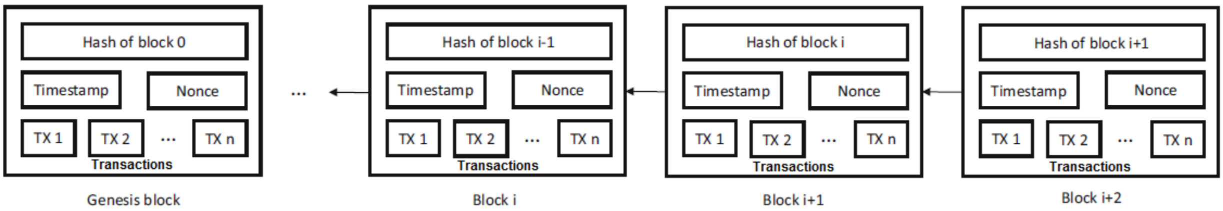


Figure 1. Basic blockchain structure.

In addition to the transactions, each block holds a Unix time timestamp which indicates the time of each transaction. Moreover, each block has a 256-bit hash value of the previous parent block and a nonce, which is a random number that verifies the hash. Several hashing algorithms can be used, but the SHA-256 cryptographic hash is the most common [7]. Hash values are unique and therefore, any alteration in the data would immediately change the respective hash value. As a result, this structure ensures the security and integrity of the entire Blockchain down to the first block known as the Genesis Block. In order to alter the contents of any block, network consensus must be achieved. This implies that a minimum of half the endpoints connected throughout the network reach a common agreement about the present state of the current distributed ledger or transaction. Hence, in large networks, Blockchain is relatively resistant to data modification due to the huge number of endpoints. On the other hand, the computational power and storage power required are immense. Various research studies around Blockchains have been published in recent years to analyze and tackle the issues and limitations in Blockchains and evaluate consensus algorithms. The practical byzantine fault tolerance algorithm (PBFT), the proof-of-work algorithm (PoW), the proof-of-stake algorithm (PoS) and the delegated proof-of-stake algorithm (DPoS) are the four main methods of reaching consensus in the current Blockchain technology. In PBFT, each node in the network maintains an ongoing internal state. When a message is received, nodes use the message in conjunction with their internal state to run a computation algorithm to validate the message. After reaching a decision about the new message, the node shares that decision with other nodes in the system. A consensus decision is determined based on the total decisions submitted by all participating nodes. This method of establishing consensus requires less effort than other methods. However, anonymity can be a great risk on the system.

An alternative method of reaching consensus within a Blockchain is the proof-of-work (PoW) algorithm, which is used by Bitcoin. In contrast to PBFT, PoW does not require all nodes on the network to participate and submit their individual conclusions in order for a consensus to be achieved. Rather, PoW is an algorithm that uses a hash function. Transactions can then be independently verified by all other system participants. The scheme allows for easy, broad participation while maintaining anonymity.

Proof-of-stake (PoS) algorithm is similar to PoW algorithm, but the participation in the consensus building process is restricted to a predefined set of nodes known for having a legitimate stake in the Blockchain. The hash function calculation is replaced with a simple digital signature which proves ownership of the stake. A more centralized way of achieving consensus is using the delegated proof-of-stake (DPoS) system. The algorithm works in a similar manner as in the PoS system, except that individuals choose an entity to represent their portions of stake in the system.

In [12], a specially designed attack scenario is presented, in which collaborating miners' revenue can be larger than their fair share. Such attacks can have significant consequences on the Blockchain structure. Miners might prefer to join the attackers and the colluding group will expand in size until it becomes a majority. At this point, the Blockchain system ceases to be a decentralized system. In [13], a decentralized smart contract system that does not store or show financial transactions is presented. Normally, all transactions, including flow of money, are exposed on the Blockchain. Such information might be useful by hackers and network attackers to track money and assets. Using such decentralized smart contract system retains transactional privacy from the public's view. In [14], a highlight of the weaknesses and limitations of Bitcoin technology is presented. This includes the theft or loss of Bitcoins due to malware attacks, structural problems and scalability issues, like delayed transaction confirmation, data retention and communication failures. A fair exchange protocol that improves the users' anonymity is used to improve the quality of the existing Bitcoin technology. In [15], a new mechanism for securing Blockchains' contracts is presented. By introducing a credibility score measure, a hybrid Blockchain that prevents an attacker from monopolizing resources is introduced. Credibility is a vital factor in any contract or transaction. Contractors must develop a good knowledge about each other to build up credibility and trust. The more contracts a contractor has with different people, the more credibility he/she gains. The mechanism proposed creates a hybrid Blockchain based on the proof-of-stake and credibility score methods. The proposed system uses the proof-of-work algorithm to introduce a hybrid solution in which power and storage requirements are minimized while improving the network's scalability and security.

Since Blockchain is a decentralized distributed ledger, it has to be managed by a peer-to-peer network adhering to a common protocol for communication and appending new blocks. Therefore, each peer in the network will have a copy of the Blockchain. Any alteration in the Blockchain will involve the alteration of all subsequent blocks, which requires consensus of the majority of the network. In its current form, the hash value of each block is generated by miners. Mining is a process at which specialized computers compete to solve a complex computational problem and produce a valid hash. Hence, miners secure the network and process every transaction. The SHA-256 cryptographic hash chooses any 256-bit number ranging from 0 to 2^{256}. The target is a 256-bit number that all Blockcahin clients share. The SHA-256 hash of a block's header must be lower than or equal to the current target for the block to be accepted by the network. The lower the target, the more difficult it is to generate a block. The maximum target defined by SHA256 mining devices is illustrated in equation 1.

$$T_{Max}= 0x00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF \quad (1)$$

We define (*D*) to be the difficulty of finding a valid hash for a given block. *D* can be defined as in Equation 2 [16]-[17]:

$$D = \frac{T_{Max}}{T_{Cur}} \quad (2)$$

where, T_{Max} is the maximum target of the hash value and T_{Cur} is the current target hash that the miner found.

Difficulty can be simply defined as the ratio between the maximum target and the current target. T_{Max} and T_{Cur} can be expressed as in Equations 3 and 4 [16]-[17]. The maximum target is defined as $(2^{16} - 1) \times 2^{208}$ or approximately 2^{224}. Since there are 2^{256} different values that a hash can take, a random hash has a chance of about 2^{-32} to be lower than the maximum target.

$$T_{Max} = (2^{16} - 1)2^{208} \cong 2^{224} \quad (3)$$

$$T_{Cur} = \frac{(2^{16}-1)2^{208}}{D} \quad (4)$$

The expected number of hashes (*N*) that a miner needs to calculate to find a block with a given difficulty (*D*) is illustrated in Equation 5 [2]:

$$N = \frac{2^{256}}{\frac{(2^{16}-1)2^{208}}{D}} = \frac{2^{256}D}{(2^{16}-1)2^{208}} = \frac{2^{48}D}{(2^{16}-1)} = \frac{2^{48}D}{(2^{16}-1)} \cong 2^{32}D \quad (5)$$

Thus, every hash produced by a given miner has a probability (*P*) to validate a given block, as seen in Equation 6:

$$P = \frac{1}{2^{32}D} \quad (6)$$

Miners from all around the world compete using powerful dedicated mining computers to generate a valid hash. From the equations above, it is evident that such process imposes high levels of power consumption. Currently, adding a single transaction to the Bitcoin Blockchain platform consumes about 600 kWh [18]-[19] and the annual power consumption for the Bitcoin network is about 77.78 TWh, which is comparable to the power consumption of Chile. Furthermore, every node in the public Blockchain network has a local copy of the entire Blockchain. This drains huge amount of storage. Currently, the Bitcoin database exceeds 250GB and is growing up in a sharp exponential rate, as seen in Figure 2. The overall Bitcoin network is consuming more than 1000 TiB of storage per year [20] and is increasing in a sharp rate. This is due to the increase in the number of users of the Bitcoin network. Such properties will restraint the use of this technology in banking and commercial businesses.

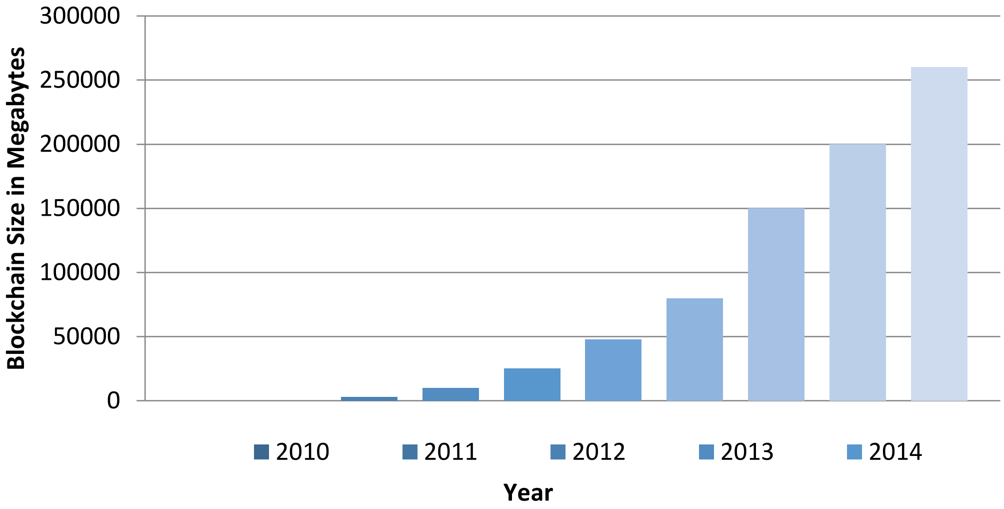


Figure 2. Size of blockchain database.

3. Proposed Hybrid Blockchain

To minimize the computational power and storage requirements, a hybrid Blockchain model that combines the centralized feature of servers with the distributed nature of Blockchains is proposed. Figure 3 illustrates the network infrastructure of the proposed system.

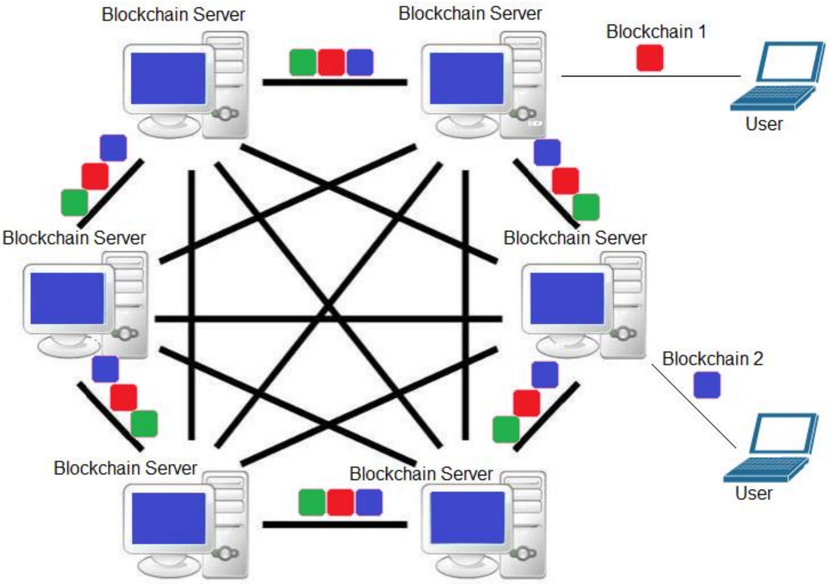


Figure 3. Proposed hybrid blockchain network infrastructure.

The proposed model can be highly beneficial to modern online businesses. Banks and e-commerce platforms have several servers distributed over a large geographical area. Using the proposed model, two sets of networks are defined: a personal Blockchain through which users can interface with the network servers and a global Blockchain consisting of interconnecting servers. The global Blockchain can be viewed as a chain of Blockchains that is distributed among servers and being updated periodically. In such scheme, users can access their respective personal Blockchain, hence accessing their private information only. No user can get hold of other users' personal data, which improves data privacy. Moreover, single or multiple private miners owned or governed by the local business policies are distributed among the network. Miners control the appending of new blocks and mapping hashes throughout the network without competing to solve a complex mathematical formula. This scenario can improve the network's security, since securing a Blockchain might require knowing participating miners' computational capabilities. This, in turn, aids in detecting potential selfish mining attacks [21-[22]. Normally, miners use the Proof-of-Work (PoW) as the security algorithm and compete trying to solve a complex computational challenge imposed by the PoW protocol. As a result, miners consume a huge amount of power [23]-[24], [15]. The proposed design includes single or multiple private trusted miners distributed throughout the network. Furthermore, the mining process can be fairly distributed among miners, hence immensely reducing the power consumption requirements of maintaining a Blockchain. Figure 4 illustrates the detailed architecture of the proposed Hybrid Blockchain.

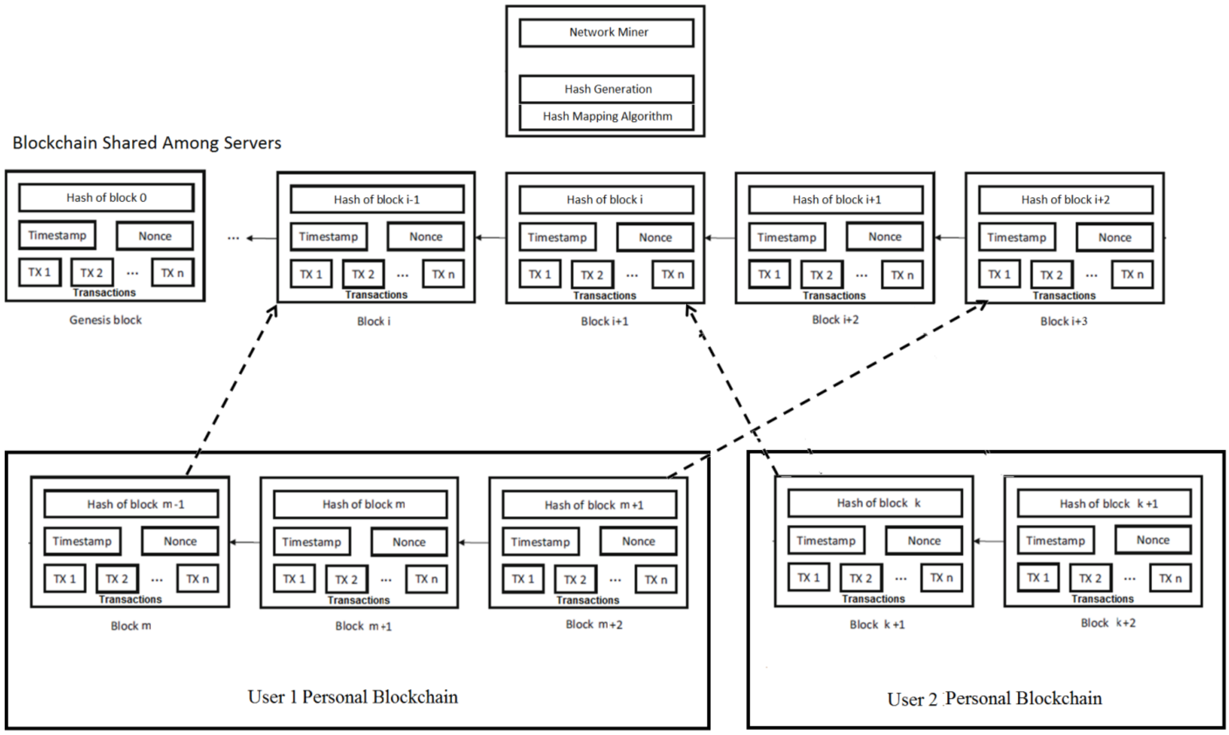


Figure 4. Proposed hybrid blockchain architecture.

In this scheme, every user has a copy of his/her personal Blockchain consisting of his/her personal transactions and information. Whenever a user executes a transaction, a broadcast request message is sent from the user's device to the miners throughout the network. Upon receiving the message, a specific miner, based on an election algorithm, will compute and produce a global hash and a private hash which is sent back to the user. Transactions and data blocks of the user's personal Blockchain are linked together through the received private hash. The user's data is saved locally on his device and also broadcasted to the data servers throughout the network. Hence, the locally saved user's Blockchain represents 50% of his/her overall personal network. The other 50% is represented by the business's servers. Each server holds a copy of a global Blockchain consisting of blocks from all connected users. The blocks are linked through a global hash generated by the network's miner. Users can only access their respective information. Therefore, no user can access or view other users' personal data, hence improving data privacy.

Since miners are responsible for the generation of private and global hashes, a mapping algorithm should be used to track any network change. Mapping algorithms, in their simplest form, can be look-up-tables that are controlled by miners to translate and govern any change in the network. Furthermore, to reduce power consumption for maintaining the Blockchain, the mining process can be fairly distributed among miners. The distribution process can be based on an election algorithm to assign a miner. The election algorithm can be chosen upon the miner's geographical area, the miner's workload, user's ping signal latency or even randomly. Figure 5 illustrates an example of an election algorithm based on a specific user's ping signal latency.

As seen in Figure 4, the network of each user is composed of the user's personal device, representing 50% of the network, along with the servers' network representing the remaining 50%. This scheme offers high levels of data security. This is due to the fact that hackers need to control the majority of the network's servers in addition to the client's device to achieve network consensus to alter or access only a single block. Assuming that there are several servers in a network, Equation 7 describes the minimal amount of devices "*K*" needed to alter a given user block, whereas Equation 8 illustrates the minimum amount of devices "*K*" needed to control several blocks within the network.

```
$R_i = Request\ signal\ from\ user\ \langle i \rangle$
$L_i = Latency\ of\ ping\ signal\ from\ miner\ to\ user\ \langle i \rangle$
$M_j = Miner\ (j)\ in\ the\ network$

$receive(\ R_i)$:
  $do$
     $ping\ (i)$
     $calculate\ (L_i)$
     $broadcast\ (M_j, L_i)$

$get(\langle M_j, (L_i) \rangle\ )_i$:
  $find\ \left(\min(M_j, (L_i)\right) \forall j$
  $assign(\ M_j)$
```

Figure 5. A sample election algorithm based on ping signal latency.

$$K \geq \frac{\Sigma s}{2} + 1 \tag{7}$$

$$K \geq \frac{\Sigma s}{2} + n \tag{8}$$

where,

S is a server holding the chain of Blockchains.
n: Number of different users within a selected portion of the Blockchain.

To reach consensus in conventional Blockchains, attackers only need to control a half of the nodes holding the data. Figure 6 illustrates the improved security of the proposed design against the current Blockchain system by comparing the minimal number of devices "*K*" needed to control the network as the number of users "*n*" increases.

Figure 7 presents the consensus algorithm used in the proposed design. Miners initiate the consensus protocol in the network through a '*proposed*' function illustrated at lines 7–14 of the algorithm, allowing them to propose new blocks. Afterwards, processes within the network decide whether a new block at a given index is valid at lines 16-23. Appending a new block is shown at line 24 depending on the function *get-main-branch( )*. Line 11 maps a global hash used in the servers' network with a local hash used within the personal network.

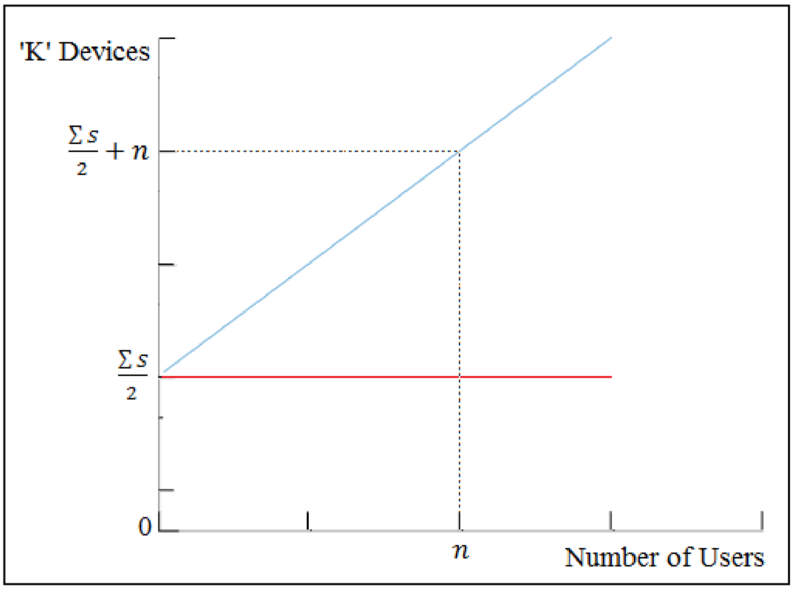


Figure 6. Number of devices to achieve network majority.

```
$j_i = \langle BL_i, PO_i \rangle$ , the local Blockchain at node $po_i$ on the servers network is a directed acyclic graph of blocks $BL_i$ and pointers $PO_i$
$\ell_i = \langle B_i, P_i \rangle$ , the local Blockchain of a given user at node $p_i$ on the user network is a directed acyclic graph of blocks $B_i$ and pointers $P_i$
$bl, b$ are blocks with the following fields
      $parent$, the block preceding $bl, b$ in the chain
      $pow$, proof − of − work nonce of $bl, b$ to verify the hash
      $children$, the blocks succeeding $bl, b$ in the chain

$proposed(\,)_i$:
  do
     $nonce = random\_number(\,)$
     create block $bl : bl.parent = last\_block(j_i)$ and $bl.pow = nonce$
     $Ln = map\_nonce(nonce)$
     create block $b : b.parent = last\_block(\ell_i)$ and $b.pow = ln$
     broadcast $(\langle\{bl\}, \{\langle bl, bl.parent\rangle\}\rangle)$
     save $(\langle\{b\}, \{\langle b, b.parent\rangle\}\rangle)$

$received(\langle B_j, P_j \rangle\,)_i$:
  $BL_i \leftarrow BL_i \cup B_j$
  $PO_i \leftarrow PO_i \cup P_j$
  $B_i \leftarrow B_i \cup B_j$
  $P_i \leftarrow P_i \cup P_j$
  $\langle \overline{BL_\iota}, \overline{PO_\iota} \rangle \leftarrow get\_main\_branch(\,)$
  $\langle \overline{\overline{B}}_\iota, \overline{\overline{P}}_\iota \rangle \leftarrow get\_user\_branch(\,)$
  if $bl_0 \in \overline{BL_\iota} \wedge \exists\, bl_1, \dots, bl_m \in BL_i : \langle bl_1, bl_0\rangle, \langle bl_2, bl_1\rangle, \dots, \langle bl_m, bl_{m-1}\rangle \in PO_i$ and
  $b_0 \in \overline{\overline{B}}_\iota \wedge \exists\, b_1, \dots, b_m \in B_i : \langle b_1, b_0\rangle, \langle b_2, b_1\rangle, \dots, \langle b_m, b_{m-1}\rangle \in P_i$ then
     $append(b_0)$
```

Figure 7. Consensus algorithm.

4. Conclusion

As centralized systems need a central owner to connect and govern all other users and devices, the system structure is highly dependent on the network connectivity. Hence, abrupt failure of the entire system due to connectivity or security issues is likely to occur. Therefore, decentralized systems emerged as an alternative solution to resolve the security issues. One of the most recent promising decentralized architectures is Blockchain technology. Ever since, Blockchain has been adopted by businesses, e-commerce platforms and digital currencies like Bitcoin. However, as Bitcoin and related crypto-currencies have become increasingly popular, they have hit scalability and reliability issues. The process of improving scalability has been obstructed by an inherent trade-off between performance metrics and security requirements of the system architecture. This paper proposes a hybrid Blockchain system that is suitable for banks and e-commerce businesses. Users can connect to servers using personal Blockchains, while servers share a chain of Blockchains throughout the business's private network. Miners governed by the business's policies control the appending of new blocks and mapping hash values. Having private miners will dramatically reduce power consumption demands and improve the overall quality of the Blockchain technology. This solution diminishes the space requirements of end users and enhances the security of the system by introducing personal networks. The proposed hybrid solution inherits the simple deployment and affordable maintenance in centralized systems while promoting resource sharing and improved scalability and fault-tolerance in decentralized systems.

References

[1] S. Nakamoto, Bitcoin: Peer-to-Peer Electronic Cash System, 2008.

[2] I. Eyal, A. Efe Gencer, E. Gün Sirer and R. van Renesse, "Bitcoin-ng: A Scalable Blockchain Protocol," Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’16), pp. 45-59, Santa Clara, CA, USA, March 16–18, 2016.

[3] L. W. Cong and Z. He, "Blockchain Disruption and Smart Contracts," The Review of Financial Studies, vol. 32, no. 5, pp. 1754-1797, 2019.

[4] R. Beck, M. Avital, M. Rossi and J. B. Thatcher, "Blockchain Technology in Business and Information Systems Research," Business & Information Systems Engineering, vol. 59, pp. 381-384, 2017.

[5] A. Wright and P. De Filippi, "Decentralized Blockchain Technology and the Rise of Lex Cryptographia," SSRN, ID. 2580664, Elsevier, 2015.

[6] G. Zyskind, O. Nathan and A. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," Proc. of IEEE Security and Privacy Workshops, pp. 180-184, San Jose, CA, USA, 2015.

[7] Z. Zheng, S. Xie, H.-N. Dai, X. Chen and H. Wang, "Blockchain Challenges and Opportunities: A Survey," International Journal of Web and Grid Services, vol. 14, no. 4, pp. 352-375, 2018.

[8] M. Nofer, P. Gomber, O. Hinz and D. Schiereck, "Blockchain," Business & Information Systems Engineering, vol. 59, no. 3, pp. 183-187, 2017.

[9] J. Lindman, V. K. Tuunainen and M. Rossi, "Opportunities and Risks of Blockchain Technologies–A Research Agenda," Proceedings of the 50th Hawaii International Conference on System Sciences, pp. 1533-1542, [Online], Available: http://hdl.handle.net/10125/41338, 2017.

[10] V. Gramoli, "From Blockchain Consensus Back to Byzantine Consensus," Future Generation Computer Systems, vol. 107, pp. 760-769, 2020.

[11] J.-H. Huh and S.-K. Kim, "The Blockchain Consensus Algorithm for Viable Management of New and Renewable Energies," Sustainability, vol. 11, no. 11, ID. 3184, [Online], Available: https://doi.org/10.3390/
su11113184, 2019.

[12] I. Eyal and E. Gün Sirer, "Majority Is Not Enough: Bitcoin Mining Is Vulnerable," Proc. of International Conference on Financial Cryptography and Data Security, pp. 436-454, Springer, Berlin, Heidelberg, 2014.

[13] A. Kosba, A. Miller, E. Shi, Z. Wen and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-preserving Smart Contracts," Proc. of IEEE Symposium on Security and Privacy (SP), pp. 839-858, San Jose, CA, USA, 2016.

[14] S. Barber, X. Boyen, E. Shi and E. Uzun, "Bitter to Better—How to Make Bitcoin a Better Currency," Proc. of the International Conference on Financial Cryptography and Data Security, pp. 399-414, Springer, Berlin, Heidelberg, 2012.

[15] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu and J. Kishigami, "Blockchain Contract: Securing a Blockchain Applied to Smart Contracts," Proc. of the IEEE International Conference on Consumer Electronics (ICCE), pp. 467-468, 2016.

[16] D. Meshkov, A. Chepurnoy and M. Jansen, "Short Paper: Revisiting Difficulty Control for Blockchain Systems," Proc. of International Workshop on Data Privacy Management, Cryptocurrencies and Blockchain

Technology, pp. 429-436, Springer, Cham, 2017.

[17] D. Kraft, "Difficulty Control for Blockchain-based Consensus Systems," Peer-to-Peer Networking and Applications, vol. 9, no. 2, pp. 397-413, 2016.

[18] H. Vranken, "Sustainability of Bitcoin and Blockchains," Current Opinion in Environmental Sustainability, vol. 28, pp. 1-9, 2017.

[19] K. J. O'Dwyer and D. Malone, "Bitcoin Mining and Its Energy Footprint," Proc. of the 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), pp. 280-285, 2014.

[20] E. Francioni and F. Venturelli, "The Dusk Network and Blockchain Architecture," WEB3 Symposium, Amsterdam, The Netherlands, [Online], Avaialble: https://dusk.network/uploads/dusk-whitepaperv1.pdf, 2018.

[21] P. H. Gleichauf, "Blockchain Mining Using Trusted Nodes," U.S. Patent Application Publication, Pub. no. US 2018 / 0109541 A1, [Online], Available: https://patentimages.storage.googleapis.com/23/4f/69/4bbc9631
31ccbc67/US20180109541A1.pdf, May 14, 2019.

[22] A. Shariar, Md. A. Imran, P. Paul and A. Rahman, "A Decentralized Computational System Built on Blockchain for Educational Institutions," Proceedings of the International Conference on Computing Advancements, pp. 1-6, [Online], Available: https://doi.org/10.1145/3377049.3377058, 2020.

[23] M. Crosby, Nachiappan, P. Pattanayak, S. Verma and V. Kalyanaraman, "Blockchain Technology: Beyond Bitcoin," Applied Innovation Review (AIR), no.8, pp. 6-19, June 2016.

[24] BitFury Group, "Proof of Stake *versus* Proof of Work," White Paper, Version 1, pp. 2-26, [Online], available: https://pdfs.semanticscholar.org/6990/0bac4097a576414f69f1998c11089fb5bb94.pdf, September 2015.

[25] J. Spasovski and P. Eklund, "Proof of Stake Blockchain: Performance and Scalability for Groupware Communications," Proceedings of the 9th International Conference on Management of Digital EcoSystems, pp. 251-258, [Online], available: https://pure.itu.dk/portal/files/83126213/Proof_of_Stake_Blockchain_Performance_and_Scalability_for_Groupware_Communications.pdf, 2017.

ملخص البحث:

تعدّ "سلسلة الوحدات" تقنية ثورية اكتسبت شعبية واسعة منذ ظهور ما يُعرف بالنقود الخفيّة. وقد تخطى استخدام هذه التقنية العُملة الرقمية الى ميدانٍ أرْحب شمل إنترنت الأشياء، وتطبيقات الأمان، والأنظمة الذكية، على سبيل المثال لا الحصر. ومع تنامي أعداد مستخدمي تلك السلاسل، بدأت بعض السلبيات بالظهور؛ إذ إنّ تلك السلاسل تستهلك كميات زائدة من الطاقة لتخزين البيانات ومعالجتها. من ناحية أخرى، فإن الطبيعة المحدودة لتلك السلاسل من حيث قابليتها للتوسيع، بسبب المتطلبات الهائلة اللازمة لتخزين البيانات، ربما تشكل معضلة. ولتحسين الأداء الإجمالي، لا بدّ من التغلُّب على العديد من التحديات في البنية الراهنة للسلاسل موضوع هذه الدراسة.

تقدم هذه الدراسة بنيةً هجينةً لسلاسل الوحدات تجمع بين الطبيعة التوزيعية للسلاسل والسمة المركزية للخوادم. في النظام المقترح، يتصل المستخدمون عبر السلاسل الشخصية، بينما تتشارك الخوادم سلسلةً من سلاسل الوحدات لضمان سلامة النظام وأمانه. وهذا من شأنه أن يقلص من متطلبات التخزين للمستخدمين النهائيين ويحسن من قابلية الشبكات للتوسيع. ومن المؤمل أن تستفيد مؤسسات الأعمال من البنية المقترحة؛ لأنها تخلق نموذج أعمالٍ موثوقاً وقابلاً للتوسيع.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

INTEGRATING UML 2.0 ACTIVITY DIAGRAMS AND PI-CALCULUS FOR MODELING AND VERIFICATION OF SOFTWARE SYSTEMS USING TGG

Raida Elmansouri[1], Said Meghzili[1], Allaoua Chaoui[1], Aissam Belghiat[2] and Omar Hedjazi[1]

(Received: 18-Apr.-2020, Revised: 18-Jun.-2020 and 26-Jul.-2020, Accepted: 24-Aug.-2020)

ABSTRACT

This paper deals with modeling and verification of software systems by combining UML diagrams and Pi-calculus. UML 2.0 Activity diagrams are used for modeling the behavior of software systems, while Pi-calculus is used for semantic and verification purposes. More precisely, UML is a semi-formal language and so it needs formal semantics for its constructs and lacks tools for verifying properties. To this end, we propose an approach and a tool called AD2PICALC for transforming UML 2.0 Activity diagrams to Pi-calculus processes using Eclipse Xpand and TGG tools. The obtained Pi-calculus processes are then used as input for Pi-calculus tools, like MWB, to verify some properties as deadlocks, safety, determinism, termination and livelock. We illustrate our contribution through an example from the literature and verify the property of deadlock using MWB tool. The main contribution of this paper lies in the automation of the transformation approach using TGG tools.

KEYWORDS

Model-driven engineering, TGG, Xpand, UML activity diagrams, Pi-calculus, Model transformation, Graph transformation, Software systems.

1. INTRODUCTION

A critical system is a system the "failure" of which is a threat to human life, to the environment of the system or to the existence of the organization that manages it. Examples of critical systems include - among others- communication systems, embedded control systems, command and control systems and transport systems. The cost of a failure in a critical system could exceed the cost of the system itself. Nowadays, most of critical systems are computer-based. Therefore, to develop powerful and sophisticated software, the modeling and verification of such systems seem to be the best solution for such task. Specifically, modeling facilitates the understanding of their complex behavior and also simulates these systems, while verification ensures their accuracy. The combination of UML diagrams and formal methods is a very suitable approach for the development of software systems [1]-[2]. Unified Modeling Language (UML) [3] is a well-known standard notation used to model object oriented software systems. It provides methods that are structured, semi-formal and graphical for specification, but not suitable for verification and validation of software systems. UML has different kinds of structural and behavioral diagrams. Each diagram is dedicated to a description of different aspects of a complex (software) system. UML Activity diagrams are used to model easily the dynamic behavior of workflow systems. One of their main purposes is to model software processes and business processes and represent control flows between activities. On the other hand, formal methods are used in software engineering to reason about mathematical models by proving or verifying properties (e.g. deadlock) of models. They are used to ensure that these systems are developed without error; i.e., these systems are free of deadlock, safe, deterministic, terminating and free of livelock. However, the analysis and design work with formal methods is very expensive and requires mathematical skills. To bridge the gap between formal methods and semi-formal ones [4], several researchers proposed approaches allowing the integration of formal models supporting formal verification in semi-formal models.

1. R. Elmansouri, S. Meghzili, A. Chaoui and O. Hedjazi are with MISC Laboratory, Department of Computer Science and Its Applications, Faculty of NTIC, University Constantine 2 - Abdelhamid Mehri, Constantine, Algeria. Emails: raida.elmansouri@univ-constantine2.dz, meghzili.said.1989@gmail.com, allaoua.chaoui@univ-constantine2.dz and hedjazi@mjustice.dz

2. A. Belghiat is with University of Jijel, Algeria. Email: belghiatissam@gmail.com

In the present work, our main contribution consists of an integrated approach and a tool (called AD2PICALC) combining UML 2.0 activity diagrams and Pi-calculus [5]-[6] for the development of software systems. This approach is based on modeling, meta-modeling and model transformation, which are the fundamental concepts of Model Driven Engineering (MDE) [7]. Indeed, MDE is an active research area in both academia and industry. It aims to decrease the complexity of software development. It allows portability, interoperability and reuse. In this paper, we propose another way and a tool for transforming UML 2.0 Activity diagrams to Pi-calculus. This approach is based on Xpand [8] and TGG tool [9] which permits a bidirectional approach. As a semi-formal notation, UML 2.0 Activity diagrams need formal semantics. So, to implement a formal analysis of a UML activity diagram specification, we propose to translate it to Pi-calculus process. Therefore, the obtained process model can be automatically verified (whether it satisfies or not certain properties, such as deadlock) using Pi-calculus analytical tools, such as MWB [10]. Pi-calculus is a simple mathematical process model based on CCS, which stands for Calculus Communicating System, a language proposed by Milner in 1980 [11]. It belongs to the family of process algebras. Since UML activity diagrams are graphs, the proposed approach is based on graph grammars. TGG is used to implement the graph grammar.

The rest of the paper is organized as follows. In Section 2, we discuss some related works. In Section 3, we recall some basic concepts about UML 2.0 activity diagrams, Pi-calculus and graph grammars. In Section 4, we propose our approach and tool that combine UML 2.0 activity diagrams and Pi-calculus process algebra for the development of software systems. In Section 5, we apply our approach on an illustrative example from the literature. Section 6 concludes the paper and gives some perspectives of this research work.

2. RELATED WORK

Many works tackled the problem of formalizing UML Activity diagram through translating it to formal standards supported by analysis facilities. In [12], the authors proposed activity diagram patterns for modeling business processes and a semantics for the activity diagrams, formalized by colored Petri nets. In [13], the authors defined semantics for activity diagram of UML by means of regular expression and its equivalent transition system. Moreover, they proposed a formal verification and traceability method for any activity model with the help of Arden's lemma. In [14], the authors transformed automatically UML 2.0 activity diagram to Petri nets. This transformation helps software designers analyze and verify properties using INA analyzer tool. In [15], the authors presented a transformation from Activity Diagram into its semantically equivalent Colored Petri Nets using Weighted Directed Graph. This transformation consists of two steps. In the first step, the UML Activity Diagram is transformed into a Weighted Directed Graph and in the second step, the Weighted Directed Graph is transformed into semantically equivalent Colored Petri Nets. In [16], the authors proposed a framework that provides formal definitions for UML Activity diagrams by transforming them to a formal representation called point graph (PG). The approach is illustrated with a case study at King’s College Hospital to improve patient flows of an accident and emergency department. In [17], the authors developed a specific tool, called MAV-UML-AD, allowing the specification and the verification of workflow models using UML activity diagrams and Event-B. The developed tool transforms an activity diagram model into an equivalent Event-B specification according to a mathematical semantics. They illustrated the use of the developed tool MAV-UML-AD using an example of specification and verification. In [18], the authors presented an approach that transforms the UML sequence diagrams, behavioural state machines and activity diagrams into a single Transition System to support model checking using NuSMV tool [19]. In [20] and [21], the authors presented an approach based on SCALA, an environment of executing Isabelle/HOL specifications that allows to transform UML State machine diagrams to Colored Petri Nets models. The authors also proved the correctness of certain structural properties of this transformation.

Several approaches proposed semantics for UML diagrams using process algebras, like Pi-calculus. In [22], the authors have proposed an automatic translation of UML specifications made up of sequence and state diagrams into Pi-calculus processes. The central point of their proposed translation was the coherence of the two types of diagrams. In [23], the authors proposed an approach for mapping only UML state machine diagrams into Pi-calculus using TGG tool. In [24], the authors presented an approach for capturing and verifying dynamic program behaviors using UML communication diagrams and Pi-calculus. In [25], the authors proposed an approach based on ATOM³ tool [26] for mapping UML 2.0 Activity diagrams to Pi-calculus. Other works used Petri nets and their extensions, like Colored Petri nets to formalize UML diagrams. In [27], the authors presented π-calculus semantics as a formal basis for UML activity diagrams. They showed manually and formally the consistency between the concepts of activity diagrams and π-calculus expressions. In our present paper, we propose an automatic mapping using TGGs tools [9] based on ideas presented in [27]. The main differences between our approach and the previously cited approaches of transforming UML diagrams to π-calculus are summarized in the following table.

	Used UML Diagrams	Manual	Automatic	Used Transformation Tool
[22]	Activity, sequence and state charts	x		
[23]	State chart		x	TGG
[24]	Communication	x		
[25]	Activity		x	ATOM^3
[27]	Activity	x		
Our Approach	Activity		x	TGG

The use of TGG graph transformation tool instead of ATOM^3 is due to its ability to perform complex transformation and offer the capabilities needed to realize our ideas. Moreover, we have applied our approach and the developed tool on a complex illustrative example.

3. BACKGROUND

In the following sub-sections, we briefly recall some basic concepts about UML 2.0 Activity diagrams, Pi-calculus and graph grammars.

3.1 UML Activity Diagrams

Activity diagram is an important UML diagram to describe dynamic aspects of a system. Activity diagram is the object-oriented equivalent of flow charts and Data Flow Diagrams (DFDs) from structured software development. It is used to represent the flow from one activity to another activity. The activity describes a particular operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams allow dealing with all types of flow control by means of different elements, like initial, flow final, activity final, decision, merge, fork and join nodes. For more details, the reader is referred to [3].

3.2 Pi-calculus

The Pi-calculus [5] is an extension of CCS [11]. It is a process algebra where processes interact by sending communication links to each other. It can be considered as a mathematical model of processes the interconnections of which change when they communicate [6]. The transfer of a communication link between two processes is the basic computational operation. Then, the link is used for further interaction with other processes. Pi-calculus offers primitives for describing and analyzing global distributed infrastructure, focusing on process migration between peer process interaction *via* dynamic channel-private channel communication. Example applications include languages supporting distributed programming with process mobility: polyphonic C#, BPML description and analysis of authentication protocols: spi calculus typed processes to ensure fine-grained resource access control. For more details, see the excellent introduction to the Pi-calculus by Joachim Parrow [6].

3.3 Graph Grammars and TGG

In the following part, we recall some concepts about graph grammars and TGG.

3.3.1 Graph Grammars

Before presenting the idea of TGGs, we begin with graph grammars [28]. A graph grammar evolves from Chomsky grammar on strings to graphs. It consists of a set of graph-rewriting rules. Each one has a graph at its Left Hand Side (LHS) and another graph at its Right Hand Side (RHS), as shown in Figure 1.

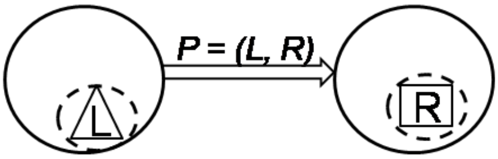


Figure 1. LHS and RHS of a rule.

The semantics of a graph grammar rule is similar to classical grammars in formal languages. A graph grammar rule can be applied to some graph called host graph. If the LHS of the rule matches a part of the host graph, this part is replaced by the RHS of the rule.

3.3.2 Triple Graph Grammars (TGGs)

Triple Graph Grammars (TGGs) have been proposed by Andy Schϋrr in 1994 for model transformation using graph grammars [28]. They allow the user to define a transformation (in both directions) in a declarative way. More precisely, Triple Graph Grammars (TGGs) are used for defining the correspondence between two different types of models *via* sets of corresponding graphs [29]. Each element of this set is a triple consisting of two independent graphs that are linked *via* a third graph, called the correspondence graph. Because of this triple structure, such a graph is also called a triple graph. These different graphs in a triple graph are typed over different type graphs. TGG rules are non-deleting graph production rules that describe how, based on a start graph or axiom, triple graphs can be created. Triple graphs that can be created by a TGG are called valid triple graphs. Transferred to the modeling world, TGGs define sets of corresponding models, also called triple models, where the independent models, called domain models, are instances of different meta-models. The domain models are linked *via* a correspondence model, which is an instance of a correspondence meta-model.

The advantages of TGGs reside in the fact that the definition can be made operational, so that one model can be transformed into the other in either direction; even more, TGGs can be used to synchronize and maintain the correspondence of the two models, even if both of them are changed independently of each other; i.e., TGGs work in an incremental way.

3.3.3 Description of a TGG Rule

It is important to notice that the models to be transformed by TGGs will be represented as object diagrams; and a class diagram represents the set of models to be considered (meta-model). So, a TGG rule consists of nodes and arcs that represent objects and links in the domain models. LHS and RHS of a rule contain nodes and arcs. The old nodes and arcs are also called context nodes (nodes products) and edges of context (arcs products). The context nodes are shown as white boxes with a black border; nodes products are shown as green boxes with a border-dark green and labeled "+ +". The arcs of context are shown as black arrows; arcs products are shown as arrows with dark green labeled ''+ +". Further constraints on attribute values and states of implementation can be formulated in a TGG rule. In a transformation, it has many strings as values of any price in the target model to be chained to different information in the source model. In a TGG rule, OCL expressions can be used in the constraints of attribute values and states of implementation. They are shown as rounded rectangles in yellow TGG rule [30].

4. OUR APPROACH

In this section, we present an approach of mapping UML 2.0 Activity diagrams to Pi-calculus expressions. The objective of this transformation is to formally verify the desired properties of models using the analytical techniques and verification tools of Pi-calculus.

The main idea of our approach is depicted in Figure 2. It consists of three steps: (1) transforming an activity diagram into its equivalent Pi-calculus model using TGG, (2) generating the Pi-calculus code from the Pi-calculus model using eclipse Xpand code generator, (3) verifying the desired properties of the target model using the MWB Tools. In the following, we present first the meta-models, next the transformation of activity diagrams to Pi-calculus processes using TGG tool and finally the translation of Pi-calculus models in abstract syntax to Pi-calculus code with Xpand tool.

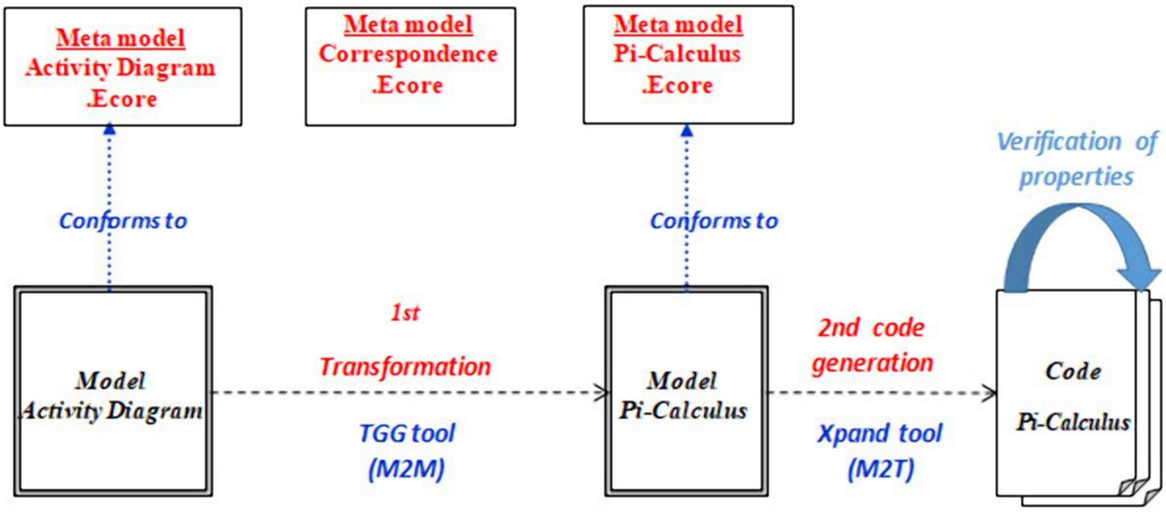


Figure 2. The architecture of our approach.

4.1 Meta-models

We follow the approach of triple graph grammars (TGGs), where the abstract syntax of both activity diagrams and Pi-calculus expressions is represented by UML object diagrams. A third meta-model is used to capture the relation between corresponding elements of activity diagrams and Pi-calculus expressions. In the following, we give the three meta-models in detail.

4.1.1 UML Activity Diagram Meta-model

We propose in Figure 3 a modified version of the meta-model of UML activity diagram (as a UML class diagram) presented in [3]. The motivation behind the modification is to adapt it to our purpose, sine the meta-model of [3] is bigger. In this meta-model, the class *ModelElement* is abstract, since a concrete element is *Activity***,** *or ActivityPartition*, *or ActivityEdge*, or *ActivityNode*. *Activity* is composed from three classes *ActivityPartition*, *ActivityEdge* and *ActivityNode*. *ActivityNode* can be *InitialNode*, *FinalNode*, *DecisionNode*, *MergeNode*, *ActionNode*, *ObjectNode*, *JoinNode or ForkNode*. *Activity Partition* is composed from *Activity Edge* and *Activity Node***;** *Activity Edge* can be *ControlFlow* or *ObjectFlow***.**

We have added to the *ControlFlow* class the following attributes:

- *visitorIN* (integer) to mark the input edges of *JoinNode* and *MergeNode***.**
- *FinIN* (boolean) to mark the last input edge of *JoinNode* and *MergeNode***.**
- *visitorOUT* (integer) to mark the output edges of *ForkNode* and *DecisionNode*.
-*FinOUT* (boolean) to mark the last output edge of *ForkNode* and *DecisionNode.*

An example of how a simple activity diagram is represented according to this meta-model is shown in Figure 4. On the left side, we find the graphical representation of an activity diagram as a concrete syntax. On the right side, we find the same pattern in its abstract syntax represented as a UML object diagram needed by TGG**.**

4.1.2 Pi-calculus Meta-model

Figure 5 shows the meta-model of the Pi-calculus processes. In this meta-model, there is a class for the root element *ProcessComposition*. This class is composed of a set of *ProcessAssignments* with at least one process. The left side of the assignment statement is the defined process (*processIdentifier*) and the right side refers to a *ProcessExpression* (process). The *ProcessExpression* can be an expression of Pi-calculus: *Prefix*, *internalchannel*, *BinaryOperation*, *Restriction* or *Empty*. A *Prefix* is composed of two parts; the left side one is an *Event* and the right side one is a *ProcessExpression*. An *Event* is: a *Silent_Event*, an *Output_Event*, an *Input_Event*, a *Condidion* or a *Concurrency*. An *Output_Event* is composed of a *Name*. An *Input_Event* consists of a *Name*. A *BinaryOperator* (*Choice*, *Concurrency or Condition)* consists of a *ProcessExpression*. A *Process* consists of a Name or more.

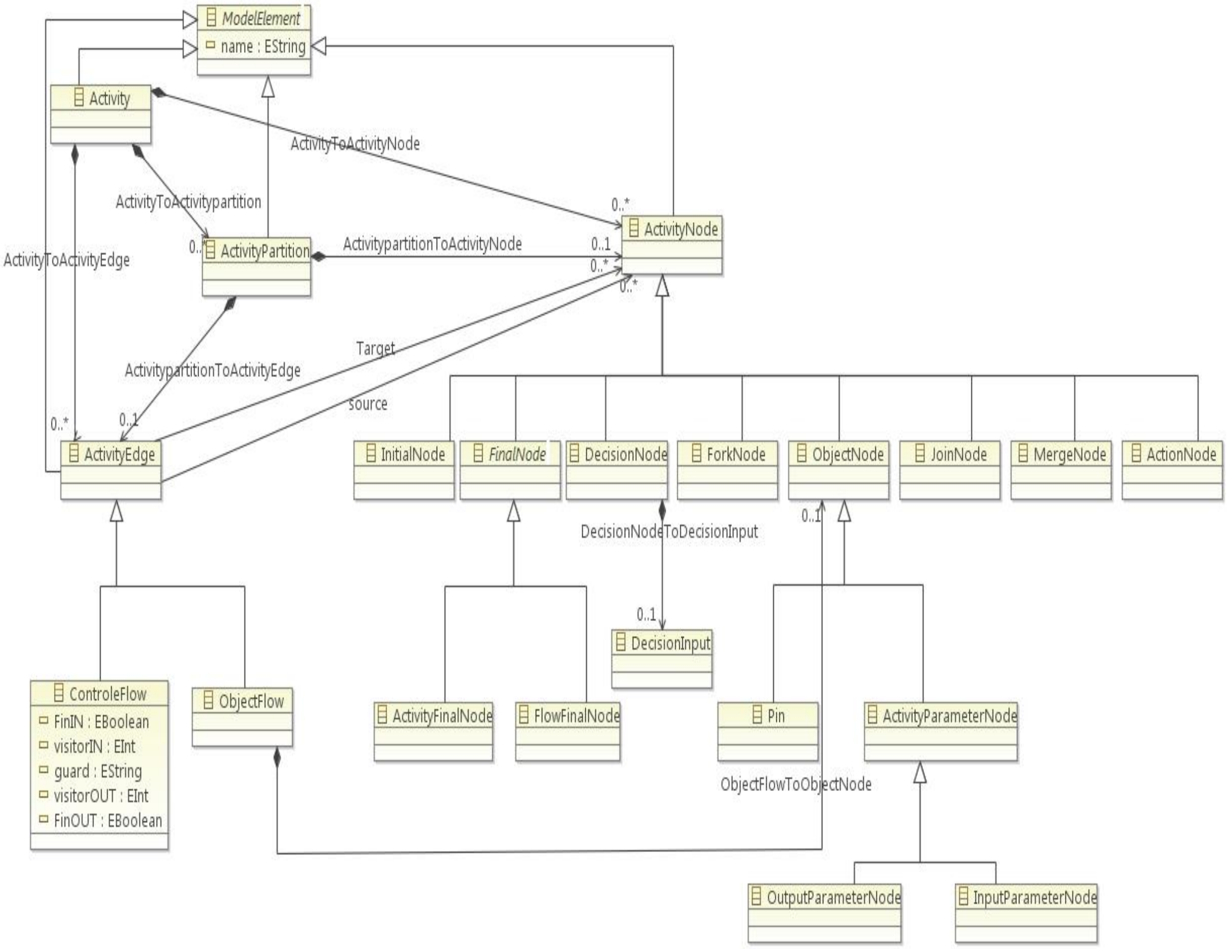


Figure 3. Adapted UML activity diagram meta-model from [3].

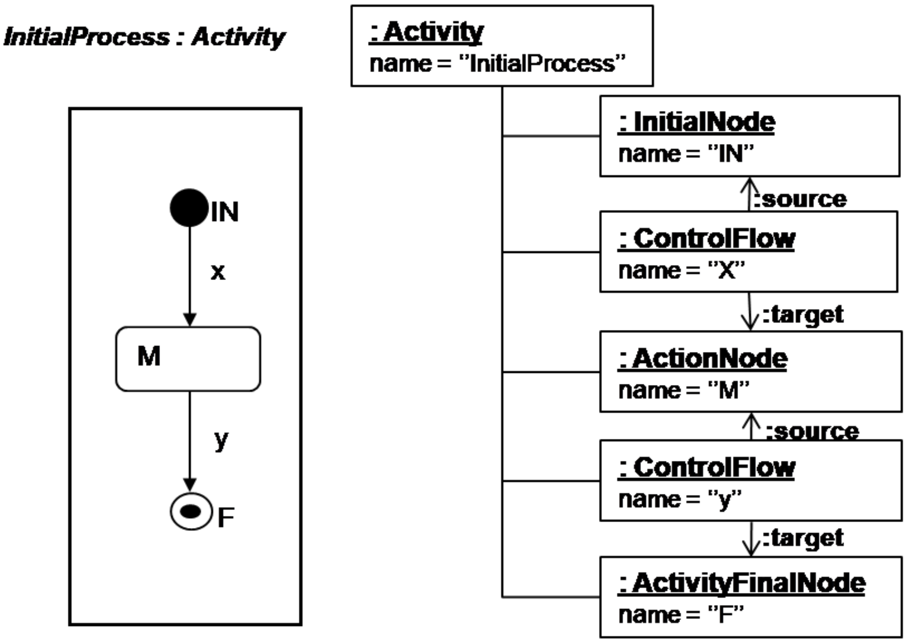


Figure 4. A simple activity diagram and its corresponding object diagram (in abstract syntax).

4.1.3 The Correspondence Meta-model

In defining the relationship between activity diagrams and Pi-calculus processes, we establish the relationship by additional nodes that refer to elements of both diagrams. These nodes are called nodes of correspondence. The exact meaning will be clearer when we define the matching rules in Section 4.2. Figure 6 shows the correspondence meta-model. We notice that these classes have other associations to classes of the two meta-models (Pi-calculus and activity diagrams), which are not shown in this diagram.

4.2 The Transformation of Activity Diagrams to Pi-calculus by a TGG Graph Grammar

In this part, we propose twenty (20) rules for transforming UML Activity diagrams to Pi-calculus processes. We recall that the source and target models are expressed as UML object diagrams. The transformation scheme is based on [27], where activity diagram nodes are transformed to Pi-calculus processes, whereas activity diagram transitions are transformed to input or output channels (names).
In the following, we give the idea of some transformation rules and their representation in TGG.

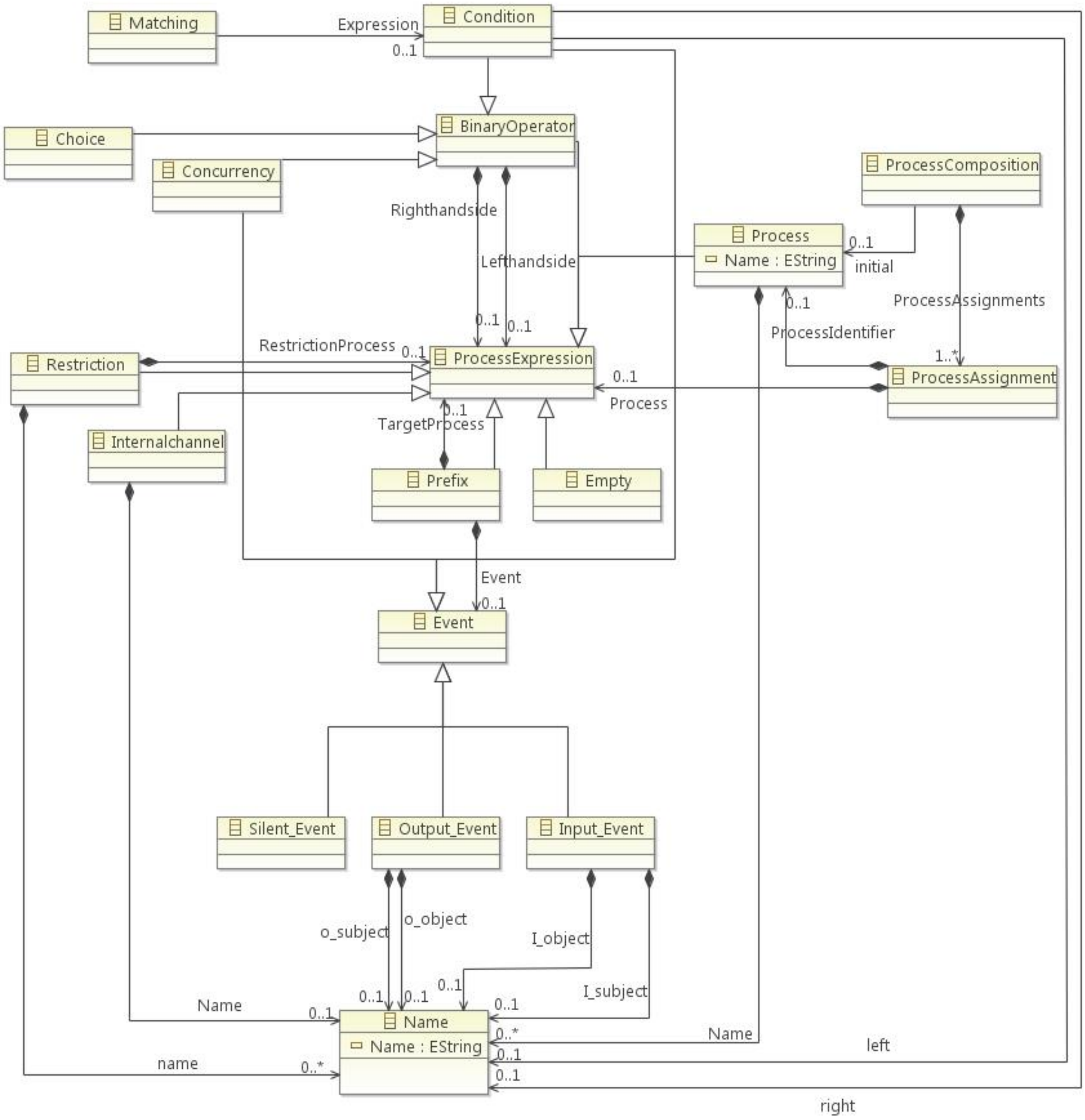


Figure 5. The Pi-calculus meta-model.

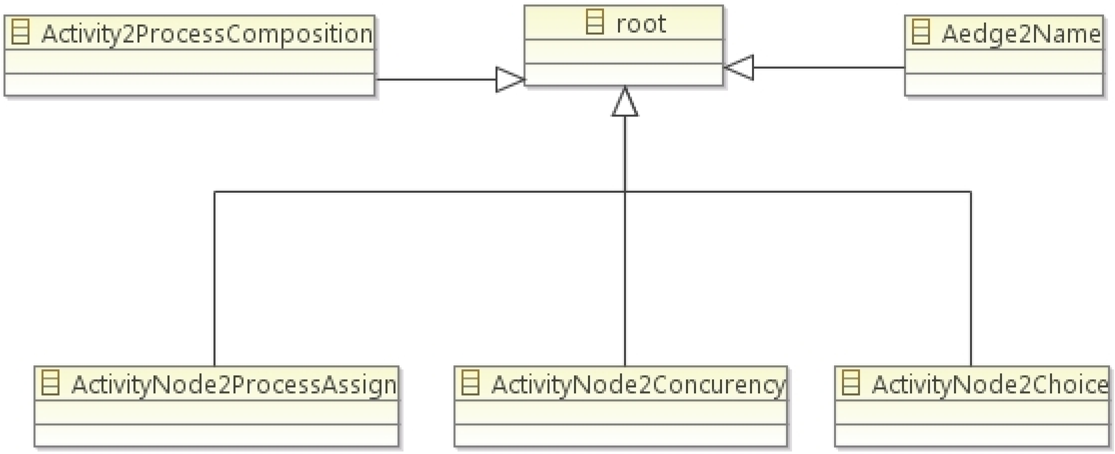


Figure 6. The correspondence model.

4.2.1 The Axiom

The axiom is the simplest relationship that transforms a simple activity diagram to Pi-calculus (Figure 7). On the right side of Figure 7, the process InitialProcess is still left open, as indicated by points then, that part will be supplemented by other rules later.

++ Activity: InitialProcess ⟺ InitialProcess (…)=(ν …) ....

Figure 7. The idea of transforming an activity diagram into an initial process in Pi-calculus.

The axiom shown in Figure 8 is the starting point of all transformations. An Activity corresponds to a *ProcessComposition*, which contains a *ProcessAssignement*. On the left side of *ProcessAssignement*, there is a *ProcessIdentifier* (Process) which takes the name of the Activity by OCL expression. On the right side, there is a *Process* (Restriction). Every pair of object diagrams for Activity diagrams and Pi-calculus that can be constructed by applying the graph grammar rules, starting from this axiom at any matching position, represents a legal relation between the two kinds of models. This is the semantics of a set of TGG-rules. From this axiom, we will now discuss the new construction occurring in the activity diagram and show how the corresponding states are created in the Pi-calculus.

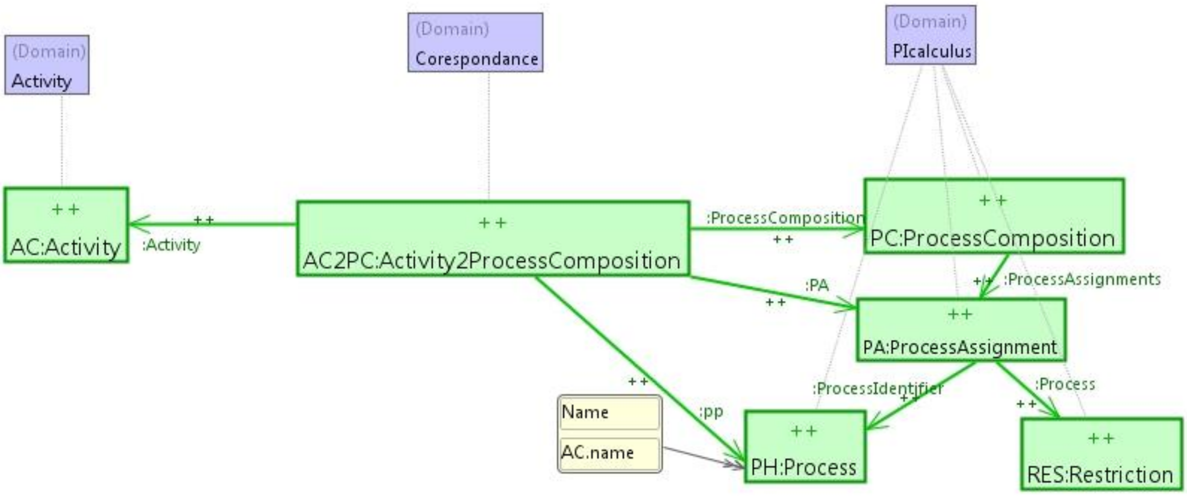
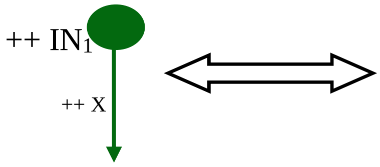


Figure 8. The axiom in TGG.

4.2.2 TGG Rule Transforming an Initial Node (InitialNode)

The idea of transforming an initial node is shown in Figure 9. At the left side, there is an initial node IN_1 connected to a control flow (indicated by green color and label-quests by + +). At the right side is the corresponding Pi-calculus code.



$IN_1\,(InStart, x) = \;\; InStart.\overline{x}.\; IN_1\,(InStart, x)$

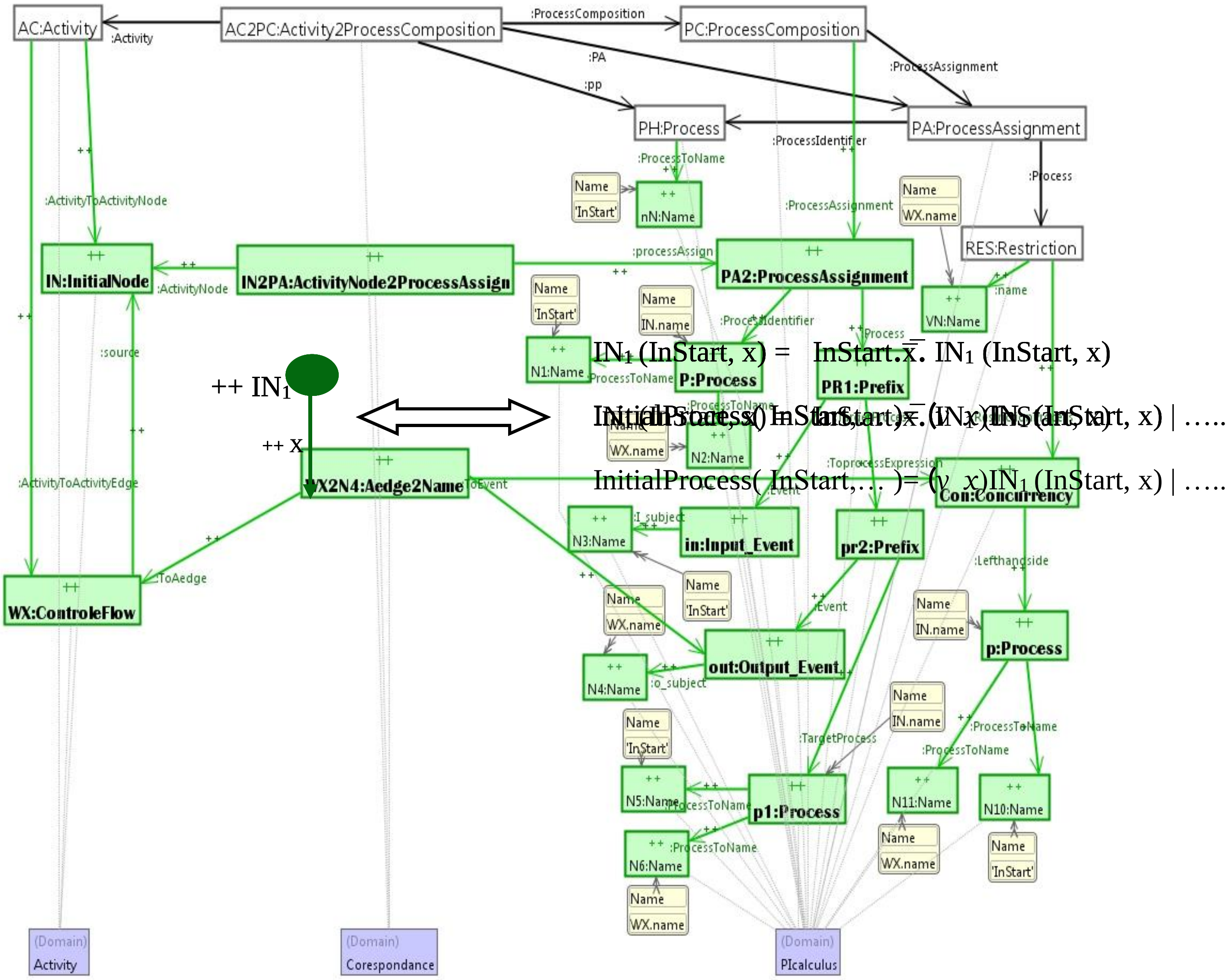
$InitialProcess(\; InStart, \ldots\;) = (\nu\;\; x) IN_1\,(InStart, x) \mid \ldots..$

Figure 9. The idea of transforming the initial node into Pi-calculus in concrete syntax.

This idea of transforming an initial node is expressed by the TGG rule in Figure 10.

4.2.3 TGG Rule Transforming an Action Node (*ActionNode*)

The idea of transforming an action node is shown in Figure 11. We assume that the control flow *x* exists and now an action *A* and a control flow *y* are added to the diagram (shown in green and marked with ++). Now, at the right side, a *ProcessAssignment* is composed of two parts: a right side *Process-*



++ IN$_1$
++ x
$IN_1\,(InStart, x) = \;\; InStart.\overline{x}.\; IN_1\,(InStart, x)$
$InitialProcess(\; InStart, \ldots\;) = (\nu\;\; x) IN_1\,(InStart, x) \mid \ldots..$

Figure 10. TGG rule transforming an initial node.

Identifier A (x, y) and a left side *ProcessExpression* containing a *Prefix* process. For the *InitialProcess*, we add a local port *y*, an operator of concurrency (|) and the process *A (x, y)*. The corresponding TGG rule is shown in Figure 12.

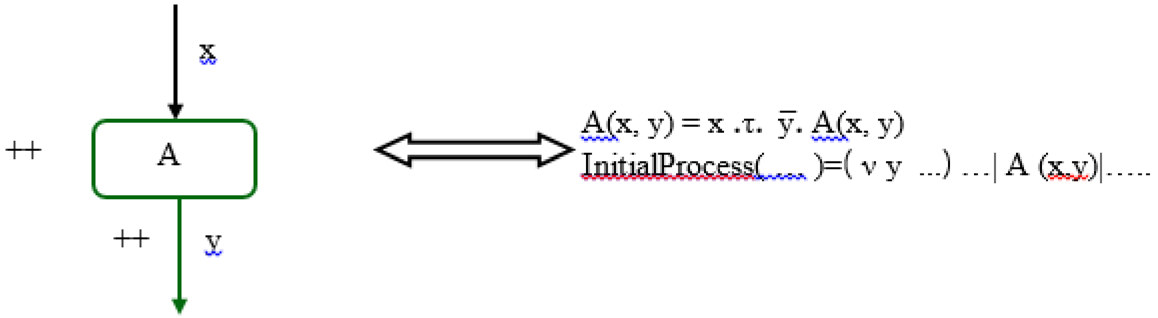

x
++
A
++
y
A(x, y) = x .τ. ȳ. A(x, y)
InitialProcess(....)=(ν y ...) ...| A (x,y)|.....

Figure 11. The action node between two control flows.

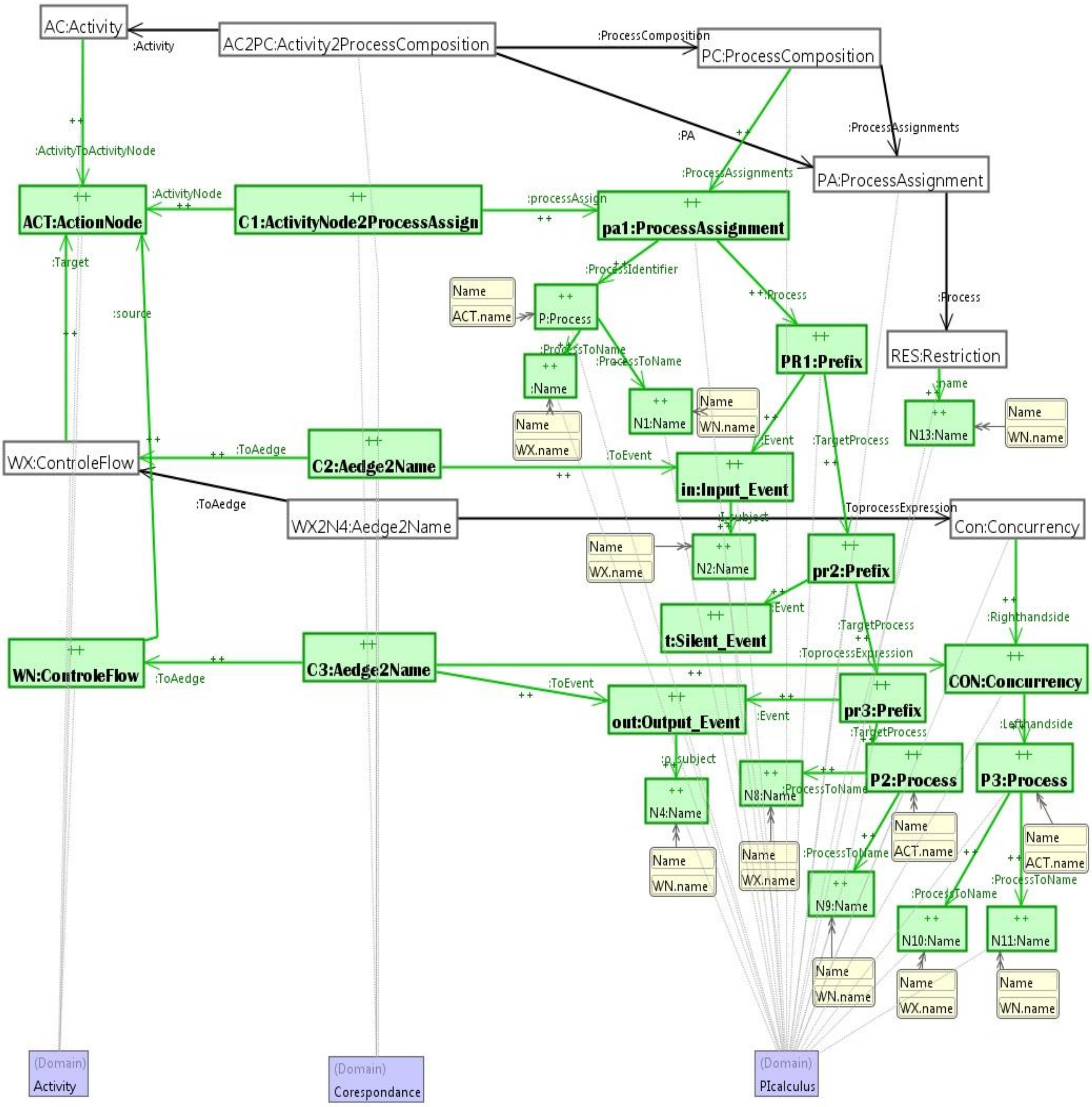


Figure 12. TGG rule transforming an Action node.

4.2.4 TGG Rule Transforming a Final Node (ActivityFinalNode)

The idea of transforming a final node is shown in Figure 13. We assume that the controle flow *y* exists and now a final node *FNA* is added to the diagram (again shown by green color and marked with ++).

A new *ProcessAssignment* is generated which consists of two parts: the *ProcessIdentifier* AFN (y) at the right side and the *prefix* process *y.AFN (y)* at the left side. In the process *InitialProcess*, we add a local port *y* as well as the *Process* AFN (y). The corresponding TGG rule is shown in Figure 14

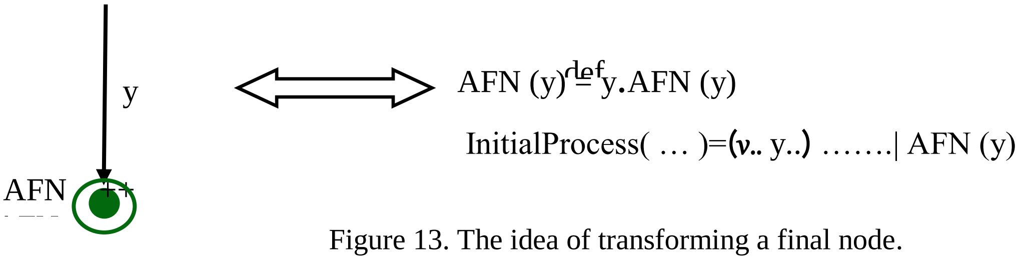


Figure 13. The idea of transforming a final node.

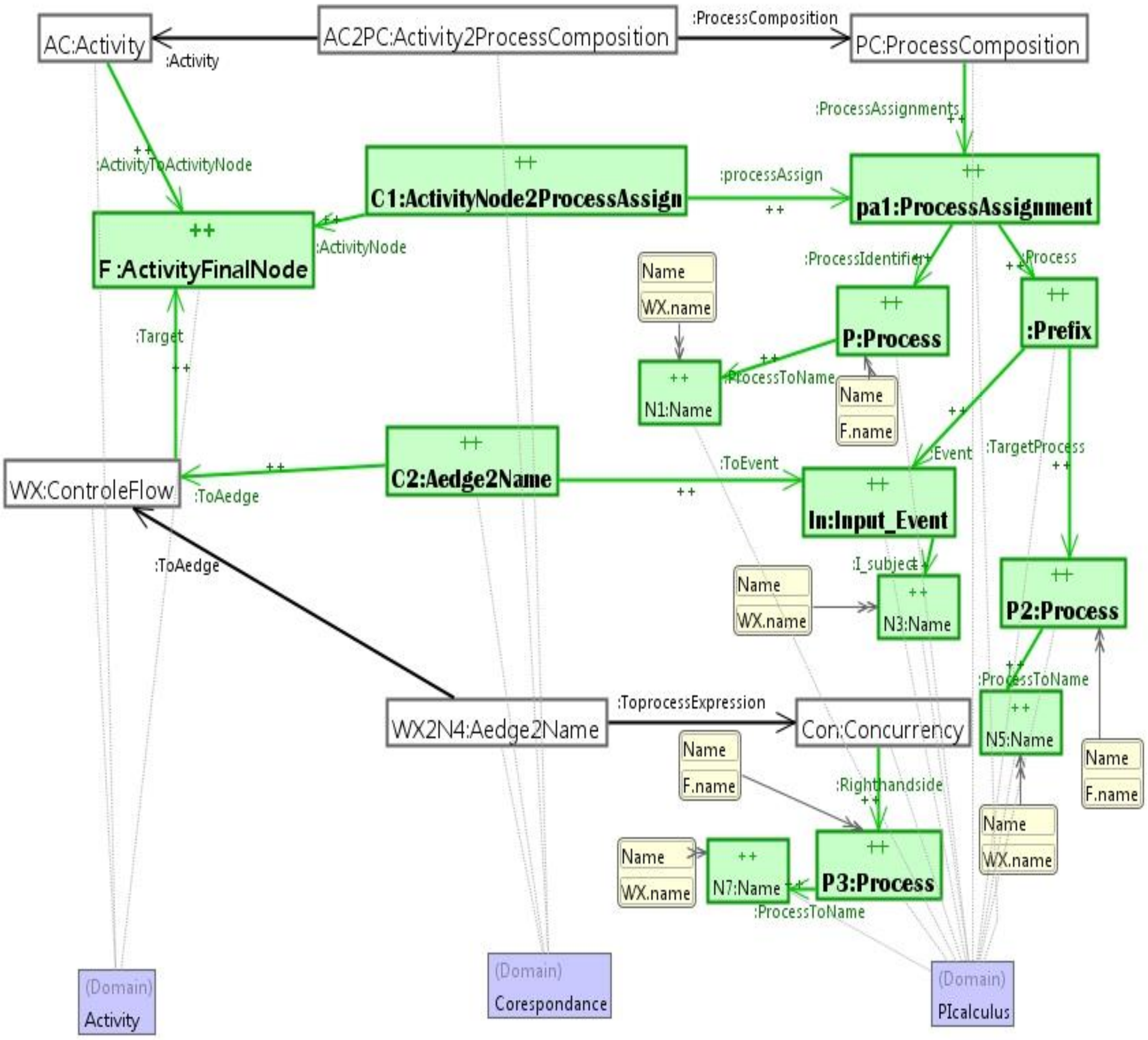


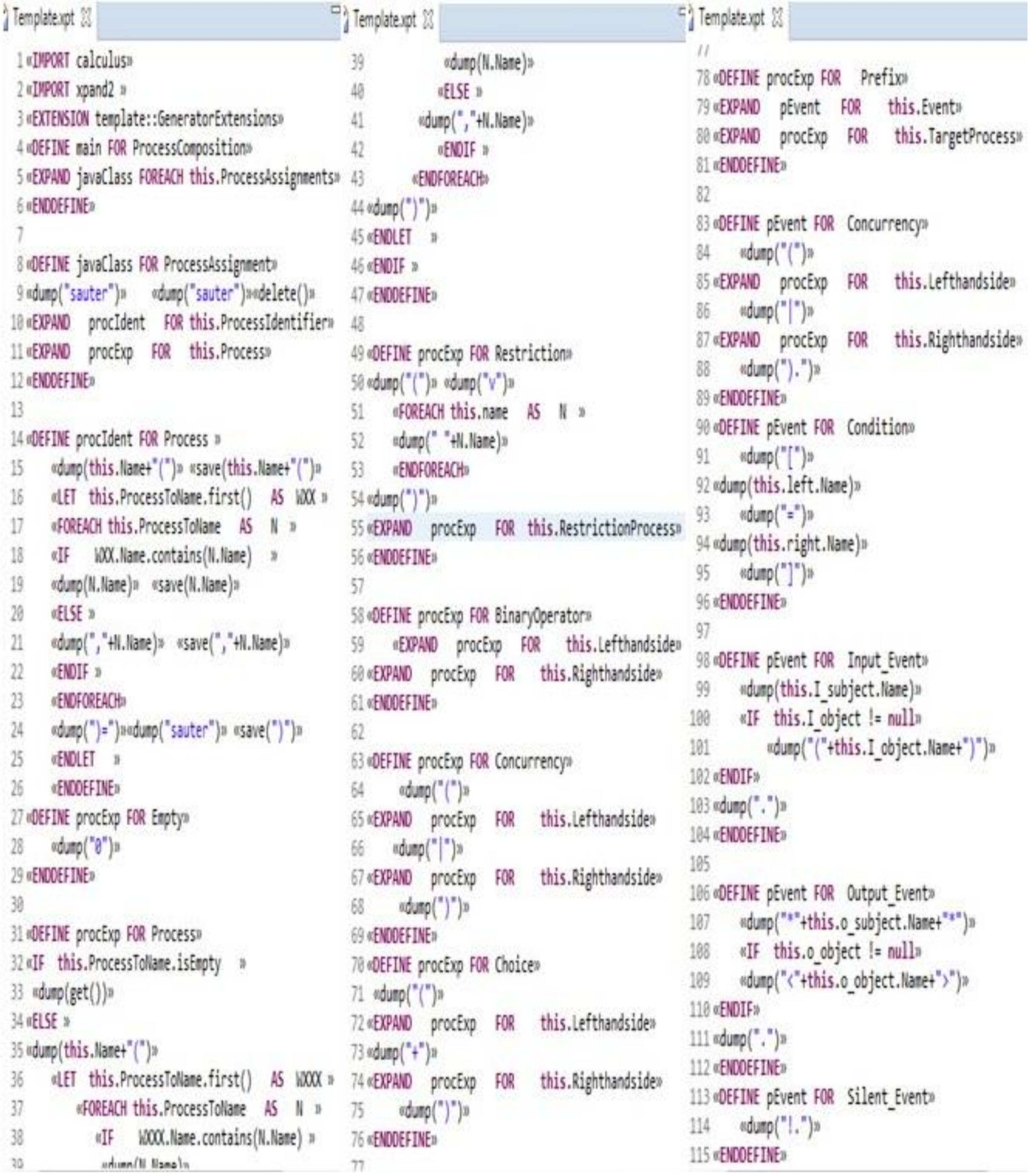
Figure 14. TGG rule for a final node.

Note: For lack of space, the reader is referred to our internal report [32] to see the rules of transforming the fusion node, the fork node, the join node and the decision node

4.3 Generating the Pi-calculus Code from the Pi-calculus Model

In order to check the correction of the target Pi-calculus models, we translate the Pi-calculus models in abstract syntax (conforming to meta-model) to Pi-calculus code (concrete syntax). This transformation is of model to text (M2T) type and is carried out using the Xpand tool of the EMF framework [31].

First, we use the Ecore meta-model of the source Pi-calculus shown in Figure 5. Second, we define the Xpand template shown in Figure 15. This template maps a Pi-calculus model in abstract syntax to Pi-calculas code.



```
«IMPORT calculus»
«IMPORT xpand2 »
«EXTENSION template::GeneratorExtensions»
«DEFINE main FOR ProcessComposition»
«EXPAND javaClass FOREACH this.ProcessAssignments»
«ENDDEFINE»

«DEFINE javaClass FOR ProcessAssignment»
«dump("sauter")»    «dump("sauter")»«delete()»
«EXPAND   procIdent   FOR this.ProcessIdentifier»
«EXPAND   procExp   FOR   this.Process»
«ENDDEFINE»

«DEFINE procIdent FOR Process »
    «dump(this.Name+"(")» «save(this.Name+"(")»
    «LET  this.ProcessToName.first()   AS  WXX »
    «FOREACH this.ProcessToName   AS   N  »
    «IF    WXX.Name.contains(N.Name)    »
    «dump(N.Name)»  «save(N.Name)»
    «ELSE »
    «dump(","+N.Name)»  «save(","+N.Name)»
    «ENDIF »
    «ENDFOREACH»
    «dump(")=")»«dump("sauter")» «save(")")»
    «ENDLET   »
    «ENDDEFINE»
«DEFINE procExp FOR Empty»
    «dump("0")»
«ENDDEFINE»

«DEFINE procExp FOR Process»
«IF  this.ProcessToName.isEmpty   »
 «dump(get())»
«ELSE »
«dump(this.Name+"(")»
    «LET  this.ProcessToName.first()   AS  WXXX »
        «FOREACH this.ProcessToName   AS   N  »
            «IF    WXXX.Name.contains(N.Name) »
                «dump(N.Name)»
              «ELSE »
           «dump(","+N.Name)»
              «ENDIF »
          «ENDFOREACH»
«dump(")")»
«ENDLET   »
«ENDIF »
«ENDDEFINE»

«DEFINE procExp FOR Restriction»
«dump("(")» «dump("v")»
    «FOREACH this.name   AS   N  »
    «dump(" "+N.Name)»
    «ENDFOREACH»
«dump(")")»
«EXPAND   procExp   FOR  this.RestrictionProcess»
«ENDDEFINE»

«DEFINE procExp FOR BinaryOperator»
    «EXPAND   procExp   FOR    this.Lefthandside»
«EXPAND   procExp   FOR    this.Righthandside»
«ENDDEFINE»

«DEFINE procExp FOR Concurrency»
     «dump("(")»
«EXPAND   procExp   FOR    this.Lefthandside»
    «dump("|")»
«EXPAND   procExp   FOR    this.Righthandside»
     «dump(")")»
«ENDDEFINE»
«DEFINE procExp FOR Choice»
 «dump("(")»
«EXPAND   procExp   FOR    this.Lefthandside»
«dump("+")»
«EXPAND   procExp   FOR    this.Righthandside»
     «dump(")")»
«ENDDEFINE»

//
«DEFINE procExp FOR   Prefix»
«EXPAND   pEvent   FOR   this.Event»
«EXPAND   procExp   FOR   this.TargetProcess»
«ENDDEFINE»

«DEFINE pEvent FOR  Concurrency»
    «dump("(")»
«EXPAND   procExp   FOR   this.Lefthandside»
    «dump("|")»
«EXPAND   procExp   FOR   this.Righthandside»
    «dump(").")»
«ENDDEFINE»
«DEFINE pEvent FOR  Condition»
    «dump("[")»
«dump(this.left.Name)»
    «dump("=")»
«dump(this.right.Name)»
    «dump("]")»
«ENDDEFINE»

«DEFINE pEvent FOR  Input_Event»
    «dump(this.I_subject.Name)»
    «IF  this.I_object != null»
         «dump("("+this.I_object.Name+")")»
«ENDIF»
«dump(".")»
«ENDDEFINE»

«DEFINE pEvent FOR  Output_Event»
    «dump("*"+this.o_subject.Name+"*")»
    «IF  this.o_object != null»
    «dump("<"+this.o_object.Name+">")»
«ENDIF»
«dump(".")»
«ENDDEFINE»
«DEFINE pEvent FOR  Silent_Event»
    «dump("!.")»
«ENDDEFINE»
```

Figure 15. The Xpand template defined to translate a Pi-calculus model to Pi-calculus code.

5. ILLUSTRATIVE EXAMPLE: TRANSFORMING AN ACTIVITY DIAGRAM TO A PI-CALCULUS EXPRESSION

We first deal with the transformation of the example. Then, we show the verification of Deadlock property using MWB Tool.

5.1 The Transformation Process

We have applied our approach on the example of Figure 16 representing an example of UML 2.0 Activity diagram borrowed from [33] with some modifications. We have first expressed this example in its abstract syntax (tree), as shown in Figure 17. Then, we have executed our graph grammar on this example using the TGG interpreter, as depicted in Figure 18.

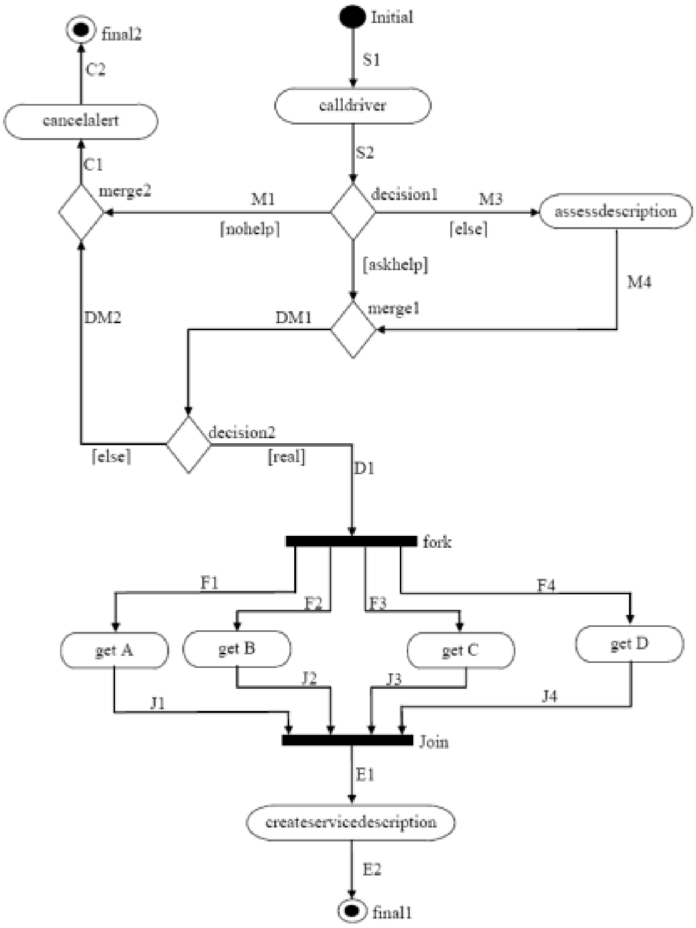
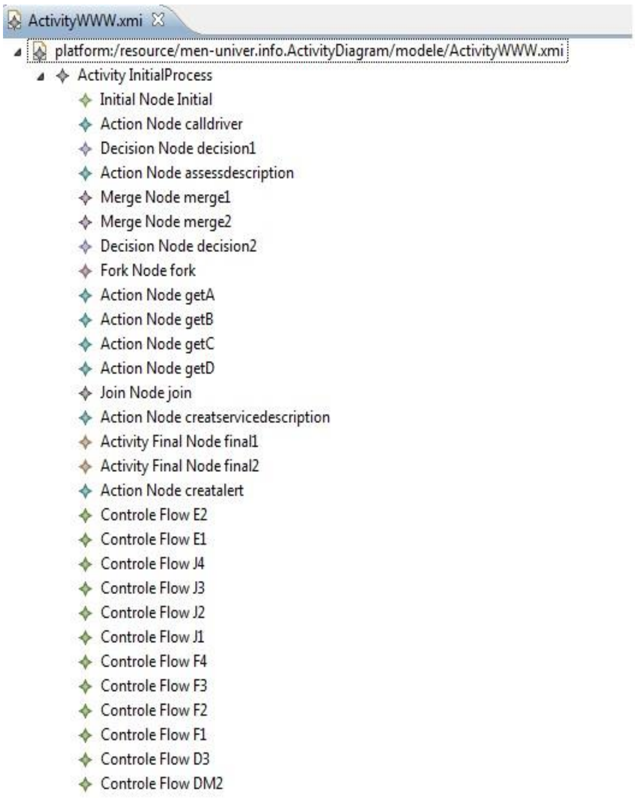
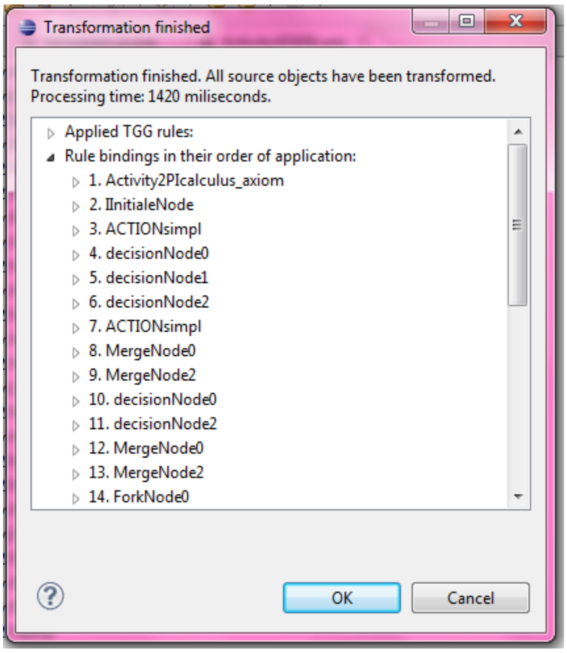


Figure 16. The activity diagram in concrete syntax.



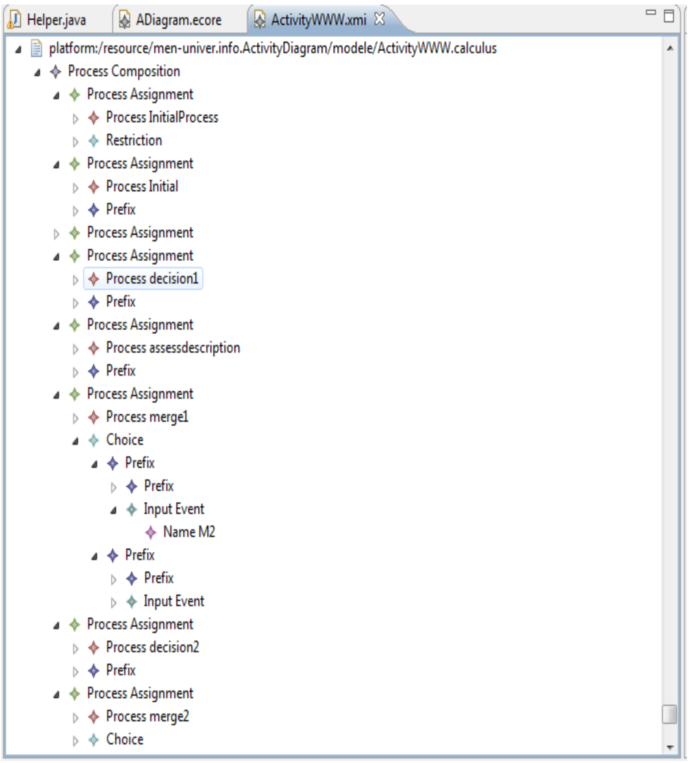
```
ActivityWWW.xmi ☒
◢ platform:/resource/men-univer.info.ActivityDiagram/modele/ActivityWWW.xmi
  ◢ ◆ Activity InitialProcess
      ◆ Initial Node Initial
      ◆ Action Node calldriver
      ◆ Decision Node decision1
      ◆ Action Node assessdescription
      ◆ Merge Node merge1
      ◆ Merge Node merge2
      ◆ Decision Node decision2
      ◆ Fork Node fork
      ◆ Action Node getA
      ◆ Action Node getB
      ◆ Action Node getC
      ◆ Action Node getD
      ◆ Join Node join
      ◆ Action Node creatservicedescription
      ◆ Activity Final Node final1
      ◆ Activity Final Node final2
      ◆ Action Node creatalert
      ◆ Controle Flow E2
      ◆ Controle Flow E1
      ◆ Controle Flow J4
      ◆ Controle Flow J3
      ◆ Controle Flow J2
      ◆ Controle Flow J1
      ◆ Controle Flow F4
      ◆ Controle Flow F3
      ◆ Controle Flow F2
      ◆ Controle Flow F1
      ◆ Controle Flow D3
      ◆ Controle Flow DM2
```

Figure 17. The activity diagram example in its abstract syntax (tree).



```
Transformation finished

Transformation finished. All source objects have been transformed.
Processing time: 1420 miliseconds.

▷ Applied TGG rules:
◢ Rule bindings in their order of application:
   ▷ 1. Activity2PIcalculus_axiom
   ▷ 2. IInitialeNode
   ▷ 3. ACTIONsimpl
   ▷ 4. decisionNode0
   ▷ 5. decisionNode1
   ▷ 6. decisionNode2
   ▷ 7. ACTIONsimpl
   ▷ 8. MergeNode0
   ▷ 9. MergeNode2
   ▷ 10. decisionNode0
   ▷ 11. decisionNode2
   ▷ 12. MergeNode0
   ▷ 13. MergeNode2
   ▷ 14. ForkNode0

? [ OK ] [ Cancel ]
```

Figure 18. Transforming the example using TGG interpreter.



```
Helper.java   ADiagram.ecore   ActivityWWW.xmi ☒
◢ platform:/resource/men-univer.info.ActivityDiagram/modele/ActivityWWW.calculus
  ◢ ◆ Process Composition
    ◢ ◆ Process Assignment
      ▷ ◆ Process InitialProcess
      ▷ ◆ Restriction
    ◢ ◆ Process Assignment
      ▷ ◆ Process Initial
      ▷ ◆ Prefix
    ▷ ◆ Process Assignment
    ◢ ◆ Process Assignment
      ▷ ◆ Process decision1
      ▷ ◆ Prefix
    ◢ ◆ Process Assignment
      ▷ ◆ Process assessdescription
      ▷ ◆ Prefix
    ◢ ◆ Process Assignment
      ▷ ◆ Process merge1
      ◢ ◆ Choice
        ◢ ◆ Prefix
          ▷ ◆ Prefix
          ◢ ◆ Input Event
              ◆ Name M2
        ◢ ◆ Prefix
          ▷ ◆ Prefix
          ▷ ◆ Input Event
    ◢ ◆ Process Assignment
      ▷ ◆ Process decision2
      ▷ ◆ Prefix
    ◢ ◆ Process Assignment
      ▷ ◆ Process merge2
      ▷ ◆ Choice
```

Figure 19. The corresponding Pi-calculus of the activity diagram example in abstract syntax.

As a result, we have obtained the Pi-calculus model (abstract syntax tree) shown in Figure 19.

Then, we have used **Xpand** tool to transform the Pi-calculus model (abstract syntax as a tree) to its textual form and obtained the final Pi-calculus code shown in Figure 20.

<u>**Notation:**</u> The generated symbols of Pi-calculus are as follows:
RESTRICTION: represented by the alphabet 'v'
OUT_EVENT: represented by the channel name + _BAR   example: x = x_bar
SILENT_EVENT: represented by 'Tau'

```
InitialProcess(InStart,nohelp,askhelp,else,else,real)=
( v  S1 S2 M1 M2 M3 M4 DM1 DM2 D3 C1 F1 F2 F3 F4 J4 J3 J2 J1 C2 E1 E2)(Initial(InStart,S1)
|(calldriver(S1,S2)|(((decision1(M1,S2,nohelp,askhelp,M2,M3,else)
|(assessdescription(M3,M4)|0))|(merge1(M2,DM1,M4)|((decision2(DM2,DM1,else,D3,
real)|((((fork(F1,D3,F2,F3,F4)|(getD(F4,J4)|0))|(getC(F3,J3)|0))|(getB(F2,J2)|0))|(getA(F1,J1)
|(join(E1,J1,J2,J3,J4)|(creatservicedescription(E1,E2)|final1(E2)))))))|(merge2(DM2,C1,M1)
|(creatalert(C1,C2)|final2(C2)))))))|0)))

Initial(InStart,S1)= InStart.S1_BAR.Initial(InStart,S1)

calldriver(S2,S1)= S1.Tau.S2_BAR.calldriver(S1,S2)

decision1(M1,S2,nohelp,M2,askhelp,else,M3)= S2.( v  X)Cn_BAR<X>.X(Y).(
[Y=nohelp]M1_BAR. decision1(M1,S2,nohelp,M2,askhelp,else,M3)+([Y=askhelp]M2_BAR. decision1(M1,S2,nohelp,M2,askhelp,else,M3)+
[Y=else]M3_BAR. decision1(M1,S2,nohelp,M2,askhelp,else,M3)))

assessdescription(M4,M3)= M3.Tau.M4_BAR.assessdescription(M3,M4)

merge1(M2,DM1,M4) = (M2.DM1_BAR. merge1(M2,DM1,M4)+M4.DM1_BAR. merge1(M2,DM1,M4))

decision2(DM2,DM1,else,real,D3)= DM1.( v  X)Cn_BAR<X>.X(Y).( [Y=else]DM2_BAR. decision2(DM2,DM1,else,real,D3)+
[Y=real]D3_BAR. decision2(DM2,DM1,else,real,D3))

merge2(DM2,C1,M1)= (DM2.C1_BAR. merge2(DM2,C1,M1)+M1.C1_BAR. merge2(DM2,C1,M1))

fork(F1,D3,F2,F3,F4)= D3.( v     traversed)(F1_BAR.traversed_BAR.0|(F2_BAR.traversed_BAR.0|
(F3_BAR.traversed_BAR.0|F4_BAR.traversed_BAR.0))).fork(F1,D3,F2,F3,F4)

getD(J4,F4)= F4.Tau.J4_BAR.getD(F4,J4)

getC(J3,F3)=  F3.Tau.J3_BAR.getC(F3,J3)

getB(J2,F2)=  F2.Tau.J2_BAR.getB(F2,J2)

getA(J1,F1)=  F1.Tau.J1_BAR.getA(F1,J1)

creatalert(C2,C1)= C1.Tau.C2_BAR.creatalert(C1,C2)

final2(C2)= C2.final2(C2)

join(J1,E1,J2,J3,J4)=
( v  received)(J1.received_BAR.0|(J2.received_BAR.0| (J3.received_BAR.0|J4.received_BAR.0))).E1_BAR.join(J1,E1,J2,J3,J4)
creatservicedescription(E2,E1)=E1.Tau.E2_BAR.creatservicedescription(E1,E2)

final1(E2)= E2.final1(E2)
```

Figure 20. The final Pi-calculus expression.

5.2 Verification of Deadlock Property

In order to verify the deadlock property, we have first installed the Microsoft windows version of MWB Tool Version 4.137 under **Standard ML of New Jersey (**SML/NJ) Version 110.57 [34]. We will verify in the following the deadlock property on two examples: the illustrative example and another example with a deadlock.

5.2.1 The Illustrative Example: with No Deadlock

The following subset has been taken as an input file process3.txt to MWB tool. We have adapted the pi-calculus expression of Figure 20 to the syntax of MWB, as shown in Figure 21.

```
agent Initialp(i,n,a,e,e10,r,c3,c4)= (^s1,s2,m1,m2,m3,m4,dm1,dm2,d3,c1,f1,f2,f3,f4,j4,j3,j2,j1,c2,e1,e2)(Initial(i,s1)  \
|(Calldriver(s1,s2)|(((Decision1(m1,s2,n,a,m2,m3,e,c3) |(Assessdescription(m3,m4)|0))|(Merge1(m2,dm1,m4)|  \
((Decision2(dm2,dm1,e10,r,d3,c4) | (((Fork(f1,d3,f2,f3)|(Getd(f4,j4)|0)|(Getc(f3,j3)|0))|(Getb(f2,j2)|0))| \
(Geta(f1,j1)|(Join(e1,j1,j2,j3,j4)|(Creatservicedescription(e1,e2)| \
Final1(e2))))))|(Merge2(dm2,c1,m1)|(Creatalert(c1,c2)|Final2(c2))))))|0)))

agent Initial(i,s1)= i.'s1.Initial(i,s1)
agent Calldriver(s1,s2)= s1.'s2.Calldriver(s1,s2)
agent Decision1(m1,s2,n,a,m2,m3,e,c3)= s2.(^x)'c3<x>.x(y).([y=n]'m1.Decision1(m1,s2,n,a,m2,m3,e,c3)+ \
([y=a]'m2.Decision1(m1,s2,n,a,m2,m3,e,c3)+ [y=e]'m3.Decision1(m1,s2,n,a,m2,m3,e,c3))) \

agent Assessdescription(m3,m4)= m3.'m4.Assessdescription(m3,m4)

agent Merge1(m2,dm1,m4)= (m2.'dm1.Merge1(m2,dm1,m4)+m4. 'dm1.Merge1(m2,dm1,m4))

agent Decision2(dm2,dm1,e10,r,d3,c4)= dm1.(^x)'c4<x>.x(y).([y=e10]'dm2.Decision2(dm2,dm1,e10,r,d3,c4)+
[y=r]'d3.Decision2(dm2,dm1,e10,r,d3,c4))

agent Merge2(dm2,c1,m1)=(dm2.'c1.Merge2(dm2,c1,m1)+m1.'c1.Merge2(dm2,c1,m1))

agent Fork(f1,d3,f2,f3,f4)=
d3.(^traversed)('f1.'traversed.0|'f2.'traversed.0|'f3.'traversed.0|'f4.'traversed.0|traversed.traversed.traversed.traversed.Fork(f1,d3,f2,f3,f4))
agent Getd(f4,j4)= f4.'j4.Getd(f4,j4)

agent Getc(f3,j3)= f3.'j3.Getc(f3,j3)

agent Getb(f2,j2)= f2.'j2.Getb(f2,j2)

agent Geta(f1,j1)= f1.'j1.Geta(f1,j1)

agent Creatalert(c1,c2)= c1.'c2.Creatalert(c1,c2)

agent Final2(c2)= c2.Final2(c2)

agent Join(e1,j1,j2,j3,j4)=
(^received)(j1.'received.0|j2.'received.0|j3.'received.0|j4.'received.0|received.received.received.received.'e1.Join(e1,j1,j2,j3,j4))
agent Creatservicedescription(e1,e2)=e1.'e2.Creatservicedescription(e1,e2)

agent Final1(e2)= e2.Final1(e2)
```

Figure 21. The final Pi-calculus expression respecting the syntax of MWB.

This code is written in the file test14.txt.

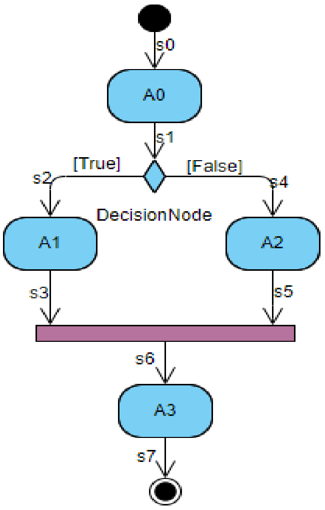
We have applied the command input "test14.txt" on MWB followed by the command deadlock Name of the agent as follows:

```
F:\sml nj 110.57\mwb99-sources>sml @SMLload=mwb $*

 The Mobility Workbench
 (MWB'99, version 4.137, built Fri Jul 24 18:10:22 2020)

MWB>input "test14.txt"
MWB>deadlocks Geta
No deadlocks found.
MWB>deadlocks Getb
No deadlocks found.
MWB>deadlocks Getc
No deadlocks found.
MWB>deadlocks Fork
No deadlocks found.
MWB>deadlocks Join
No deadlocks found.
MWB>deadlocks Final1
No deadlocks found.
MWB>deadlocks Final2
No deadlocks found.
......
MWB>
```

In conclusion, all the agents do not contain deadlock.
In the following, we show an example containing a deadlock.

5.2.2 Example of UML-AD with the Presence of Deadlock

The UML Activity diagram shown in Figure 22 is used to illustrate the presence of a deadlock.



```
Agent Initialp(i,t1,f,c)=(^s0,s1,s2,s3,s4,s5,s6,s7)(Initial(i,s0)|A0(s0,s1)|
Decision1(s1,s2,s4,t1,f,c)|\
A1(s2,s3)|A2(s4,s5)|Join(s3,s5,s6)|A3(s6,s7)|Final(s7))

agent Initial(i,s0)= i.'s0.Initial(i,s0)
agent A0(s0,s1)= s0.'s1.A0(s0,s1)
agent Decision1(s1,s2,s4,t1,f,c)=
s2.(^x)'c<x>.x(y).([y=f]'s1.Decision1(s1,s2,s4,t1,f,c)+ \
[y=t1]'s4.Decision1(s1,s2,s4,t1,f,c))
agent A1(s2,s3)= s2.'s3.A1(s2,s3)
agent A2(s4,s5)= s4.'s5.A2(s4,s5)
agent Join(s3,s5,s6)=
(^received)(s3.'received.0|s5.'received.0|received.received.'s6.Join(s3,s5,s6))
agent A3(s6,s7)= s6.'s7.A3(s6,s7)
agent Final(s7)= s7.Final(s7)
```

Figure 23. The Pi-calculus code equivalent to the UML-AD of Figure 22.

Figure 22. Example of AD with deadlock presence.

First, we have transformed this UML-AD to its equivalent Pi-Calculus code, as shown in Figure 23. Then, we have executed the command ***input "deadlock1.txt"*** **under MWB, followed by the command *deadlocks Initialp*, as follows:**

```
F:\sml nj 110.57\mwb99-sources>sml @SMLload=mwb $*

The Mobility Workbench
 (MWB'99, version 4.137, built Fri Jul 24 18:10:22 2020)

MWB>input "deadlock1.txt"
MWB>deadlocks Initialp
Deadlock found in (^~v,~v6,~v7,~v8,~v9,~v10,~v11,~v12)('~v.Initial<i,~v> | '~v6.A0<~v,~v6> |
~v7.(^x)'c<x>.x(y).([y=f]'~v6.Decision1<~v6,~v7,~v9,t1,f,c> + [y=t1]'~v9.Decision1<~v6,~v7,~v9,t1,f,c>) | ~v7.'~v8.A1<~v7,~v8> |
~v9.'~v10.A2<~v9,~v10> | (^received)(~v8.'received.0 | ~v10.'received.0 | received.received.'~v11.Join<~v8,~v10,~v11>) |
~v11.'~v12.A3<~v11,~v12> | ~v12.Final<~v12>)
 reachable by 3 commitments
MWB>
```

The response is that there is a deadlock. The interpretation of this deadlock is that the corresponding activity diagram has a design error. The result of the decision node is true or false. So, the join node will never be executed.

6. Conclusion

This paper is a contribution in the area of model-driven engineering; it is essentially based on the combined use of meta-modeling and model transformation. We have proposed an integrated approach, supported by a tool called AD2PICALC, which combines UML 2.0 Activity diagrams and Pi-calculus process algebra for the development of software systems. More precisely, we have proposed an automated approach for transforming UML 2.0 Activity diagrams to Pi-calculus processes using Eclipse Xpand and TGG tools. First, we have proposed three meta-models; one for activity diagrams, the second for Pi-calculus and another one for correspondence. Second, we have presented the first transformation (TGG rule graph grammar) from UML activity diagram to Pi-calculus models using TGG tool. Finally, we have defined the second transformation that generates the Pi-calculus code from the Pi-calculus models (abstract syntax) using Xpand tool. We have illustrated our approach through an example from the literature. In a future work, we plan to apply our approach on several real case studies and use the Pi-calculus tools, such as MWB, to verify other properties of the modeled system, such as safety, non-determinism, termination and liveness. We plan also to transform other UML diagrams, like overview interaction diagrams. Finally, we plan to deal with the verification of the transformation itself based on the work published in [26].

ACKNOWLEDGEMENTS

This work is supported by MISC laboratory, Faculty of NTIC, University Constantine 2-Abdelhamid Mehri, Constantine, Algeria and DGRSDT, Ministry of Higher Education and Scientific Research, Algeria.

REFERENCES

[1] M. Singh, A. K. Sharma and R. Saxena, "An UML+ Z Framework for Validating and Verifying the Static Aspect of Safety Critical System," Procedia-Computer Science, vol. 85, pp. 352-361, 2016.

[2] R. M. Borges and A. C. Mota, "Integrating UML and Formal Methods," Electronic Notes in Theoretical Computer Science, vol. 184, pp. 97-112, 2007.

[3] OMG UML, [Online], Available: http://www.omg.org/spec/UML/2.2/Superstructure.

[4] F. R. Golra, F. Dagnat, J. Souquières, I. Sayar and S. Guerin, "Bridging the Gap Between Informal Requirements and Formal Specifications Using Model Federation," Proc. of the 16th International Conference on Software Engineering and Formal Methods (SEFM 2018), pp. 54-69, Toulouse, France, Jun. 2018,

[5] R. Milner, Communicating and Mobile Systems: The Pi Calculus, Cambridge University Press, 1999.

[6] J. Parrow, "An Introduction to the Pi-calculus," Chapter to Appear in Handbook of Process Algebra, Ed. Bergstra, Ponse and Smolka, Elsevier, [Online], Available: http://courses.cs.vt.edu/cs5204/fall09-kafura/Papers/PICalculus/Pi-calculus-Introduction.pdf.

[7] A. R. Da Silva, "Model-driven Engineering: A Survey Supported by the Unified Conceptual Model," Computer Languages, Systems & Structures, vol. 43, pp. 139-155, 2015.

[8] S. Efftinge, P. Friese, A. Hase, D. Hübner, C. Kadura, B. Kolb et al., "Xpand Documentation," Technical Report, Copyright 2004-2010, [Online], Available: https://git.eclipse.org/c/m2t/org.eclipse.xpand.git/plain/doc/org.eclipse.xpand.doc/manual/xpand_reference.pdf, 2004.

[9] TGG Home Page, [Online], Available: www.informatik.uni-marburg.de/~swtagtive-contest/.

[10] B. Victor and F. Moller, "The Mobility Workbench—A Tool for the π-calculus," Proc. of the International Conference on Computer Aided Verification, pp. 428-440, Springer, Berlin, Heidelberg, 1994.

[11] R. Milner, "A Calculus of Communicating Systems," Lecture Notes in Computer Science, vol. 92, 1980.

[12] É. André, C. Choppy and G. Reggio, "Activity Diagram Patterns for Modeling Business Processes," Software Engineering Research, Management and Applications, Part of Studies in Computational Intelligence, vol. 496, pp. 197-213, Springer, Heidelberg, 2014.

[13] B. Hazela, D. Arora and V. Saxena, "Formalizing Semantics for UML Activity Diagram through Regular Expression Translation," Research Journal of Applied Sciences, Engineering and Technology, vol. 11, no. 2, pp 169-175, 2015.

[14] Y. Rahmoune, A. Chaoui and E. Kerkouche, "A Framework for Modeling and Analysis of UML Activity Diagram Using Graph Transformation," Procedia-Computer Science, vol. 56, pp. 612-617, 2015.

[15] M. Jamal and N. A. Zafar, "Transformation of Activity Diagrams into Colored Petri Nets Using Weighted Directed Graph," Proc. of the IEEE International Conference on Frontiers of Information Technology (FIT), pp. 181-186, Islamabad, Pakistan, December 2016.

[16] I. Chishti, A. Basukoski, T. Chaussalet and N. Beeknoo, "Transformation of UML Activity Diagram for Enhanced Reasoning," Proceedings of the Future Technologies Conference, pp. 466-482, Springer, Cham, 2018.

[17] A. Achouri, Y. B. Hlaoui and L. J. B. Ayed, "Institution-based UML Activity Diagram Transformation with Semantic Preservation," International Journal of Computational Science and Engineering, vol. 18, no. 3, pp. 240-251, 2019.

[18] L. B. R. dos Santos, V. A. de Santiago Junior and N. L. Vijaykumar, "Transformation of UML Behavioral Diagrams to Support Software Model Checking," Proc. of Formal Engineering Approaches

to Software Components and Architectures (FESCA), vol. 147, pp. 133–142, doi:10.4204/EPTCS.147.10, 2014.

[19] NuSMV Home Page, "NuSMV: A New Symbolic Model Checker," [Online], Available: http://nusmv.fbk.eu/.

[20] S. Meghzili, A. Chaoui, M. Strecker and E. Kerkouche, "On the Verification of UML State Machine Diagrams to Colored Petri Nets Transformation Using Isabelle/HOL," Proc. of IEEE International Conference on Information Reuse and Integration (IRI), pp. 419-426, San Diego, CA, USA, 2017.

[21] S. Meghzili, A. Chaoui, M. Strecker and E. Kerkouche, "Verification of Model Transformations Using Isabelle/HOL and Scala," Information Systems Frontiers, vol. 21, no. 1, pp. 45-65, 2019.

[22] K. Pokozy-Korenblat and C. Priami, "Toward Extracting π-calculus from UML Sequence and State Diagrams," Electronic Notes in Theoretical Computer Science, vol. 101, pp. 51-72, 2004.

[23] A. Belghiat and A. Chaoui, "A TGG Approach for Bidirectional Automatic Mapping between UML and Pi-calculus," Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication, Article no. 70, pp. 1-3, [Online], Available: https://doi.org/10.1145/2816839.2816857, ACM, 2015.

[24] A. Belghiat, A. Chaoui and M. Beldjehem, "Capturing and Verifying Dynamic Program Behaviour Using UML Communication Diagrams and Pi-calculus," Proc. of IEEE International Conference on Information Reuse and Integration, pp. 318-325, San Francisco, CA, USA, 2015.

[25] A. Belghiat and A. Chaoui, "A Graph Transformation of Activity Diagrams into π-calculus for Verification Purpose," Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering (ICAASE18), pp. 107-114, Constantine, Algeria, Dec. 2018.

[26] MSDL, "AToM^3 Quick Links," [Online], Available: http://atom3.cs.mcgill.ca/.

[27] V. S. Lam, "On π-calculus Semantics As a Formal Basis for UML Activity Diagrams," International Journal of Software Engineering and Knowledge Engineering, vol. 18, no. 4, pp. 541-567, 2008.

[28] A. Schürr, "Specification of Graph Translators with Triple Graph Grammars," Proc. of the International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, vol. 903, pp. 151-163, Springer, Berlin, Heidelberg, 1994.

[29] A. Königs, "Model Transformation with Triple Graph Grammars," Proc. of Model Transformations in

Practice Satellite Workshop of MODELS, A., no. 166, pp. 1-16, [Online], Available: https://pdfs.semanticscholar.org/f608/40351e4f18c6513465956361b99a0eabb148.pdf, 2005.

[30] J. Greenyer and J. Rieke, "Applying Advanced TGG Concepts for a Complex Transformation of Sequence Diagram Specifications to Timed Game Automata," Proc. of the International Symposium on Applications of Graph Transformations with Industrial Relevance, pp. 222-237, Springer, Berlin, Heidelberg, 2011.

[31] F. Budinsky, D. Steinberg, E. Ellersick, T. J. Grose and E. Merks, Eclipse Modeling Framework: A Developer's Guide, Addison-Wesley Professional, 2004.

[32] R. Elmansouri, S. Meghzili, A. Chaoui, A. Belghiat and O. Hedjazi, "Transformation Rules of UML Activity Diagrams to Pi-calculus Using TGG," Internal Report, MFGL Team, MISC Laboratory, University Constantine 2-Abdelhamid Mehri, Algeria, [Online], Available: https://misc-lab.org/en/teams/show/MFGL.

[33] D. Bisztray, K. Ehrig and R. Heckel, "Case Study: UML to CSP Transformation," Applications of Graph Transformation with Industrial Relevance, [Online], Available: https://www.informatik.uni-marburg.de/~swt/agtive-contest/UML-to-CSP.pdf, 2007.

[34] SMLNJ, "Standard ML of New Jersey," [Online], Available: http://smlnj.org.

ملخص البحث:

تُعالج هذه الورقة البحثية نمذجة أنظمة البرمجيات والتحقق منها عن طريق دمج مخططات (UML) وحساب التفاضل والتكامل (PI). وتستخدم مخططات الفعاليات (UML 2.0) لنمذجة سلوك أنظمة البرمجيات، في حين يستخدم حساب التفاضل والتكامل (PI) لأغراض تتعلق بدلالات الألفاظ والتحقق. وبشكلٍ أدقّ, فإنّ (UML) هي لغة شبه منهجية، وبذلك تحتاج لنظامٍ منهجي فيما يتعلق بدلالات الألفاظ لتركيباتها، كما تنقصها الأدوات اللازمة للتحقق من خصائصها.

من أجل ذلك، نقترح منهجاً وأداة تسمى (AD2PICALC) لتحويل مخططات الفعاليات (UML 2.0) الى عمليات تفاضل وتكامل باستخدام أدوات (Eclipse Xpand) و (TGG). ومن ثم تُستخدم العمليات الناتجة كمدخل لأدوات حساب التفاضل والتكامل، مثل MWB؛ للتحقق من عدد من الخصائص، مثل: حالات التوقف التام، والسلامة، والحتمية، والإنهاء، والتثبيت. ومن ناحية أخرى، تم عرض مساهمتنا من خلال مثال من الأدبيات ذات العلاقة، بالإضافة الى التحقق من خاصية التوقف التام باستخدام أداة (MWB). وتكمن المساهمة الأساسية لهذا البحث في أتمتة طريقة التحويل باستخدام أدوات قواعد الرسوم البيانية الثلاثية (TGG).

IMPROVING RESPONSE TIME OF TASK OFFLOADING BY RANDOM FOREST, EXTRA-TREES AND ADABOOST CLASSIFIERS IN MOBILE FOG COMPUTING

Elham Darbanian, Dadmehr Rahbari, Roghayeh Ghanizadeh and Mohsen Nickray

(Received: 27-May-2020, Revised: 7-Aug.-2020, Accepted: 26-Aug.-2020)

ABSTRACT

The application of computing resources through mobile devices (MDs) is called Mobile Computing; between cloud datacentres and devices, it is known as (Mobile) Fog Computing (MFC). We ran Cloudsim simulator to offload tasks in suitable Fog Devices (FDs), cloud or mobile. We stored the outputs of the simulator as a dataset with features and a target class. A target class is a device in which tasks are offloaded and features of tasks are authentication, confidentiality, integrity, availability, capacity, speed and cost. Decision Tree (DT), Random Forest (RF), Extra-trees and AdaBoost classifiers were classified based on attribute values and the plot of trees was drawn. According to the plot of these classifiers, we extracted each sequential condition from root to leaves and inserted it into the simulator. What these classifiers do is to improve the conditions that should be inserted in the corresponding section of the simulator. We improved the response time of offloading by Random Forest, Extra-trees and AdaBoost over Decision Tree.

KEYWORDS

Fog computing, Decision tree classifier, Random forest classifier, Extra-trees classifier, AdaBoost classifier, Offloading, Machine learning.

1. INTRODUCTION

The application of computing resources through mobile devices (MDs) is called Mobile Computing. The Mobile Computing environment includes four properties of mobility; diversity of network access types, frequent network disconnection, poor reliability and poor security [1]. Between cloud datacentres and Internet of Things (IoT) devices is (Mobile) Fog Computing nods. MFC acts as an intermediate layer between IoT devices/sensors and cloud datacentres. So, it is closer to the IoT devices to handle real-time services and latency-sensitive services and provide better Quality of Service (QoS). Routers, switches, set top boxes, proxy servers, Base Stations (BS), …etc. are in the Fog Computing environment. They can support application execution [2].

Mobile Edge Computing provides services and computing capabilities at the edge of the mobile network and can optimize existing mobile infrastructure services. MEC servers are deployed at multiple locations at the edge of the mobile network to implement the MEC environment [3]. Mobile Cloud Computing extends the computing capabilities to constrained resource mobile devices and benefits from a combination of different technologies (e.g. service-oriented computing, virtualization and grid computing). Mobile devices, communication technology and cloud servers are three main portions of it. Storage, processing, computing and security mechanism for mobile devices are provided by a cloud server through communication technologies [4]. In Table 1, the differences between the three methods including Mobile Cloud Computing, (Mobile) Edge Computing and (Mobile) Fog Computing were presented [5].

Decision Tree can be defined as a non-parametric supervised learning method. It is trained on labeled data to classify it and an acyclic directed graph is built using top-down recursive partitioning of the dataset [6]. In this paper, the dataset has some features and labels that specify the target class. Decision Tree predicts the value of a target inferred from the data features. Iterative Dichotomiser 3 (ID3),

E. Darbanian, D. Rahbari, R. Ghanizadeh and M. Nickray are with the Department of Computer Engineering and Information Technology, University of Qom, Alghadir Ave., Qom, Iran. Emails: edarbanian@gmail.com, d.rahbari@stu.qom.ac.ir, ghanizadehroghayeh1@gmail.com and m.nickray@qom.ac.ir

C4.5, C5.0 and Classification and Regression Trees (CARTs) are various Decision Tree algorithms. We use the Decision Tree classifier code in scikit-learn website which uses an optimized version of the CART algorithm [7].

Table 1. The difference between the three methods Mobile Cloud Computing, (Mobile) Edge Computing and (Mobile) Fog Computing.

	Rich in computing and/or storage resources	Rich in energy and/or power resources	Computation is mainly at the network edge	Data storage is mainly at the network edge	Interaction with remote infrastructure (cloud)	Context (location, activity, …etc.) awareness	Supporting real-time control and interactive services	Support throughput applications	Augmenting cloud data centre services	Augmenting mobile device performance
Mobile Cloud Computing	yes	yes	no	yes	yes	no	yes	no	no	yes
(Mobile) Edge Computing	yes	no	yes	yes	no	yes	yes	no	yes	yes
(Mobile) Fog Computing	yes	yes	yes	yes	no	yes	yes	no	yes	yes

Random Forest fits 100 Decision Tree classifiers by default on the dataset's different sub-samples and improves the predictive accuracy using averaging [8]. We used the Random Forest classifier code in the scikit-learn website. It is applied to the dataset and trees are constructed while the resultant individuals are combined to predict the class label. The word random is for two reasons: first, random sampling for drawing samples and second, selecting attributes or features for generating Decision Trees randomly. AdaBoost and Bootstrapping techniques are used in Random Forest to construct multiple classifiers [9].

An Extra-trees classifier fits some randomized Decision Trees on various sub-samples of the dataset, improves the predictive accuracy and controls over-fitting implementing a meta-estimator and averaging, respectively. It is similar to the Random Forest [10].

Another classifier is AdaBoost. It is a meta-estimator that begins by fitting a classifier on the original dataset. Then, additional copies of the classifier are fitted on the same dataset. However, the weights of wrong classified instances are adjusted such that subsequent classifiers focus more on difficult cases [11]. Boosting methods train predictors consecutively and try to improve their predecessors. AdaBoost is similar to Random Forest at a high level, because it collects the predictions made by each Decision Tree within the forest. Some differences between them are in AdaBoost; the Decision Trees have a depth of 1 and the final prediction made by the model is impacted by the predictions made by each Decision Tree [12]. All these classifiers are popular tools in machine learning.

Key contributions of our paper are:

1- Each FD has its features and parameters based on its internal structure on which the best FD is chosen for the module placement. The parameters that were used in this paper are authentication, confidentiality, integrity, availability, capacity, speed and cost. In this paper, we had four FDs that were called FD1, FD2, FD3 and FD4. The Cloudsim simulator that we used assigns values between 0 and 1 to features at random. Depending on values, tasks are offloaded in suitable FDs and otherwise in cloud or mobile. We stored the outputs of the simulator as a dataset.

2- Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers classify based on feature values and draw the plot of a tree. According to the plot of these classifiers, we extracted each sequential condition from root to leaves and inserted them into the simulator in the corresponding section. This reduced the number of conditions that should be inserted in the corresponding section of the simulator and response time of offloading.

3- Since Random Forest, Extra-trees and AdaBoost classifiers consider 100 different trees by default from which we choose the best one with the highest accuracy, they had a better response time compared to that of Decision Tree.

4- In practice, some of these parameters are not used. What these classifiers do is to improve the conditions that should be inserted in the corresponding section of the simulator. These methods are suitable for tasks that require a shorter response time. In fact, in the operational environment, the values of features and target class (a device in which tasks are offloaded, such as mobile, FD or cloud) can be stored as data in a dataset. Then, by using these classifiers and dataset, we can insert the conditions effectively into the simulator for the next tasks and achieve less response time.

The rest of the paper is organized as follows. Related works are presented in Section 2. In Section 3, the system model is described. The proposed approach is provided in Section 4. In Section 5, the evaluation results of the simulation are described. At last, Section 6 presents the conclusions.

2. RELATED WORK

Task offloading has been noticed a lot in recent years. Related papers were categorized as follows:

Authors in [13] expressed Android Unikernel, a short run time designed for mobile computing offloading under MFC and MEC scenarios. It also has been argued that advanced unikernel is used as a runtime in MEC or MFC to support mobile quality. To this, the concept of Rich-Unikernel was considered which aims to support various applications in one unikernel while avoiding their time-consuming recompilation. In [14], a computation offloading problem was provided in a fog computing system, which uses fog computing to answer computation requests by validating requests through the fog node or central cloud increasing the performance of applications, such as power consumption and delay. Also, the game theory approach was used to minimize the running cost. Specifically, a Generalized Nash Equilibrium Problem (GNEP) was formulated and addressed with various constraints by using the exponential penalty function method and semi-smooth Newton method.

Another way to minimize energy consumption, delays and costs was provided in [15]. Researchers investigated the problem of power consumption, performance delays and costs in a mobile fog computing system. Here, queue theory was used to derive analytical results on power consumption, delay in performance and cost by assuming three different queuing models in MD, fog node and central cloud. Based on this analysis, the multi-part optimization problem has been formulated with a common goal of minimizing energy consumption, delay in execution and cost by optimizing the probability of optimal offloading and power transfer for each mobile device. Also, using an Interior point method-based algorithm, a multi-segment problem with various limitations has been developed.

Authors in [16] proposed container transfer algorithms and architectures to support moving tasks with different needs. Also, the container migration problem of mobile application tasks in large-scale FC was modeled. Then, container transfer algorithms to support moving tasks are proposed. This has significantly reduced latency, power consumption and transmission costs. In [17], the researchers investigated a problem of cost-based fairness in a min-max computation system by optimizing offloading and resource allocation decisions, which minimized the delay cost weighting and energy consumption in the system. To address the Np-hard problem, the computation offloading and resource allocation algorithm (CORA) was proposed, which has low complexity and the offloading decisions are taken at random.

In [18], the performance of SIMDOM (A framework for SIMD instruction translation and offloading in heterogeneous mobile architectures) framework was discussed from various dimensions, such as FMEC and MCC offloading, application partition and increasing input sizes. The SIMDOM framework was evaluated in terms of parameters, such as energy, time and performance of MFLOPS. Comparison with state-of-the-art Qemu-based compiled code-offloading framework was performed, where it was found that the SIMDOM framework provides better results. Paper [19] evaluates information about IoT programs in unstable channel conditions and suggests a new way to model the quality of unstable channels. An optimal programming model and a way to reduce the complexity of the algorithm were proposed, which improved the quality of the algorithm. Besides, an offloading data scheduling algorithm (DEED) was proposed aiming at reducing energy consumption.

In [20], a near-end network solution of computation offloading on the edge/fog of the mobile was presented. Mobility, heterogeneity and geographical distribution of mobile devices are challenges of computing offloading at the edge/fog. For consideration of the computational resource demand, an independent q-learning management framework was presented. The proposed method significantly improved computational offload discharge performance by minimizing computational delay. In [21], the various types of offloading techniques that have recently been introduced in fog-driven literature or edge computing in the cloud-IoT environment are discussed. Some criteria determine when offloading was performed. Finally, the research challenges related to offloading are highlighted in the fog calculations.

In [22], deep reinforcement learning was proposed to solve the problem of offloading large-scale multi-service nodes of MEC and multiple dependencies in mobile tasks. Then, the offloading strategy by each algorithm was simulated on the edge computing iFogSim simulator platform. At last, the advantages and disadvantages of each algorithm are evaluated by comparing different factors, including power consumption, cost, load balancing, delay and network usage. Paper [23] has modeled the expected time and energy cost for different options for offloading a task on the edge, cloud or the device itself. Authors in [24] raised the issue of offloading optimization and then used the metaheuristic method to find the best policy. Also, Simulated Annealing-based Offloading Algorithm (SAOA) has also been proposed to provide a node access estimation policy based on a variety of health care information. Sensitive requirements should be met related to the different roles in IoT for architectural and algorithm design. A blockchain-based Edge ABC architecture and a Task Offloading and Resource Allocation (TO-RA) algorithm have been proposed to meet the requirements [25].

In [26], an efficient resource allocation and computation offloading model for a multiuser MEC system was proposed. Also, Advanced Encryption Standard (AES) method has been introduced to protect sensitive information from cyber-attacks and an optimization problem has been developed for mobile users to minimize energy consumption and delay latency. In [27], effective and efficient services in the fog computing environment called for a decentralized management plan of mobile edge server with p2p activation. In [28], a task-offloading and resource-scheduling algorithm was proposed to solve the problems of minimizing energy consumption and processing time of task offloading in the MEC system.

Internet of connected vehicles (IoV) has been introduced as a technology to provide tracking information to drivers and transportation control systems [29]. In this paper, to reduce execution time and energy consumption while satisfying the privacy of computational tasks, an edge computing method called edge computing-enabled computation offloading (ECO) was presented [29]. In [30], a new offloading strategy based on the firefly technique was presented. The firefly method was designed to address the offloading strategy in the Fog-Cloud environment, which selects a suitable computational device for each application.

In [31], a smart energy management method was proposed to increase the lifetime of the network in a fog computing network. A clustering mechanism was introduced for a computation degradation scenario. In [32], the problem of task mapping and scheduling (TMS) in wireless sensor networks was investigated. Its main goals were to improve runtime, power consumption and network lifetime. The MODIFIED RANDOM BIT CLIMBING (λ -MRBC) method was used to obtain the optimal solution faster. In [33], the negotiation between publisher and fog node was formed as an optimization issue. In [34], to provide effective services for performing tasks sensitive to latency and computation, a positive decision-making algorithm and resource allocation based on deep learning have been developed to minimize time and energy consumption in fog. In [35], fog calculations were introduced in a three-tier architecture to minimize energy consumption. To minimize energy consumption, an energy consumption oriented offloading algorithm for fog computing has been suggested.

Paper [36] presented and analyzed a vector instruction offloading framework (SIMDOM) in heterogeneous compute architectures. In [37], a mathematical model is presented to facilitate the calculation of computational time and energy.

We compare the mentioned offloading and scheduling methods by objectives, network architecture, environment, advantages and disadvantages in Table 2.

Table 2. Comparison of offloading and scheduling methods mentioned.

Algorithm	**Objectives**	**Network and Environment (N and E)**	**Advantages**	**Disadvantages**
Unikernel [13]	Time, memory and energy	N: MFC E: Android- x86	Compared to when running Android VM or Android container, it has the advantages of being small, fast and secure.	- Multi-process applications are not allowed. - Those codes need to fork new processes and cannot run in Android Unikernel.
MOIPM [15]	Energy, delay and cost	N: MFC E: Simulation	Low algorithm complexity.	Not suitable for delay-sensitive applications.
GNEP [14]	Execution cost	N: MFC E: Simulation	The proposed algorithm improves performance and accuracy.	Not suitable for delay-sensitive applications.
DQLCM [16]	Computational delay and power consumption	N: MFC E: Real	Among the many deep learning reinforcement algorithms, it has the advantage of fast decision-making.	May lead to a long delay in processing the work.
CORA [17]	Energy, delay and cost	N: MFC/ MCC E: MATLAB	Unlike previous work, they emphasize influential decision-making [39], [40] or resource allocation [41], with consideration of cloud and fog in common.	Has not considered the queue length and delay of user equipment request.
SIMD [18]	Energy, MFLOPS and execution time	N: MFC/ MEC/ MCC E: Real	Ability to reload from a server and execute SIMD instructions while saving energy and reducing runtime.	The simdom framework does not provide energy efficiency for metrics that have less computation.
OFFLOADING [21]	Reducing energy consumption	N: MFC/ MEC E: Real	Offers complete classification of offloading schemes.	Failure to explain how to do the review.
DEED [19]	Reducing energy consumption	N: MFC/ MEC/ MCC E: Simulation	Providing innovative work for optimal energy consumption.	Too much interpretation
IDRQN [22]	latency and network load	N: MEC E: Simulation	Better performance in power consumption, load balance, latency and average runtime.	Locking in scalability and limited to relatively few problems.
Deep Q-learning [20]	Minimizing delay	N: MFC/ MEC E: MATLAB	Ability of parallel execution.	Lack of expression of the future work.
Modeling time and energy cost [23]	Time and energy cost	N: MEC/ MCC E: Simulation	Dynamic offloading decision-making	Edge devices should be adjacent to IoT devices and their resources are limited.
SAOA [24]	Minimizing delay	N: MFC E: MATLAB	Quick response to a user request.	Privacy may not be protected.
TO-RA [25]	Reduction of delay	N: MEC E: Simulation	Compared to other algorithms, more stable and higher user	- Limitation on computing resources and storage of smart devices.

			scarification.	- Impossibility to perform computational tasks with high complexity for a long time.
Multi-users Computation Offloading Decision [26]	Minimizing time and energy consumption	N: MEC E: MATLAB	Low delay	Increasing the number of Mus causes severe interference.
Decentralized mobile edge server management plan [27]	Minimizing runtime and reducing energy consumption	N: MEC E: simulation	The proposed method is effective and practical.	The search space of this method is very large and time-consuming.
Distributed [28]	Minimizing energy consumption and processing time	N: MEC E: simulation	First work on dynamic task offloading and resource scheduling.	Lack of expression of future work
ECO [29]	Optimizing time and reducing energy consumption	N: MEC E: simulation	Preservation privacy	The smaller the scale of the vehicle, the longer the transmission time.
Firefly [30]	Minimizing computation time and energy consumption	N: MFC/ MCC E: simulation	- Automatic split - Easily finding the best solution	It is difficult to efficiently allocate IoT applications between the fog node and the cloud datacenter.
Prediction-based Energy Harvesting Scheme and Clustering [31]	Increasing network lifetime	N: MFC E: MATLAB	Saving energy and reducing latency	Management complexity
λ -MRBC [32]	Improvement of execution time, energy consumption and network lifetime	N: Wireless E: simulation	The network lifetime is prolonged through using the proposed algorithm.	Lack of expression of future work
Design of an incentive mechanism [33]	Reducing energy consumption and delay	N: MFC E: simulation	Increasing transfer speed	There may be an asymmetry between publisher and fog node.
DLJODRA [34]	Reducing energy consumption and delay	N: MFC E: Tensorflow and MATLAB	Increasing network efficiency	Deep learning-based computation offloading scheme does not consider the optimization allocation of network resources.
Energy consumption-oriented offloading algorithm [35]	Minimizing energy consumption	N: MFC/ MCC E: MATLAB	Better performance	It would have been better if it had introduced a multi-user model.
A framework for translating pre-compiled vector instructions [36]	Saving energy and time	N: MEC/ MCC E: Real	Leading to increased translation training efficiency.	Lack of expression of future work
Mathematical model for calculating the time and energy	Reducing computational time and energy	N: MCC E: simulation	Boosting performance and energy efficiency	It would have been better if authors also talked about the energy consumption coefficients of the

consumption of application models [37]				prominent parameters for smartphones.

3. SYSTEM MODEL

The system architecture is shown in Figure 1 [41]. There are mobile devices at the lowest level. For transferring data to routers, access points or base stations are used and routers send data to the closest FDs for task processing. When the sum of memory power and CPU power consumption is less than Wi-Fi's power consumption, tasks run in mobile; otherwise, they are offloaded in FDs according to their properties. If tasks do not execute in FDs, send them finally to the highest level in the cloud.

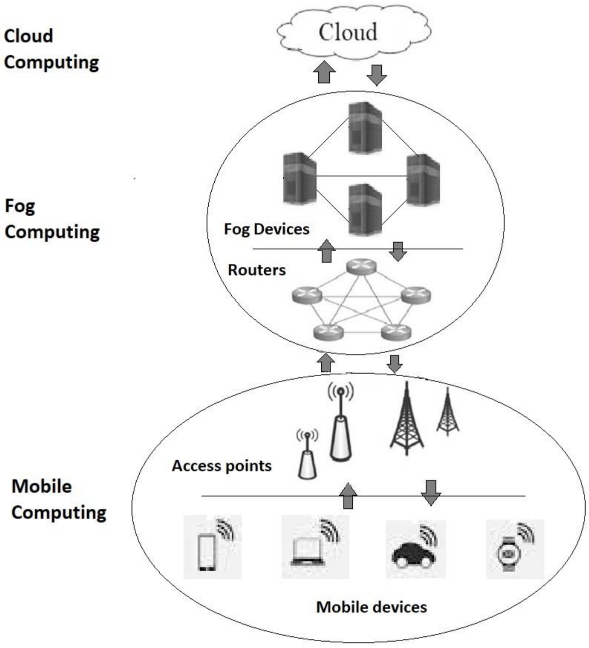


Figure 1. System architecture.

When computing is performed by mobile or portable devices (e.g. laptops, tablets or mobile phones), it is called mobile computing or nomadic computing. It is not suitable for many recent computational challenges, because of the requirements of connected consumer devices. Therefore, fog computing and cloud computing are used for more advanced calculations, because they have more resource-rich hardware [42].

3.1 Calculating Power

We used three resources for power consumption as CPU, RAM and Wi-Fi in our model, which are explained as follows [41].

3.1.1 Power Consumption of CPU

Frequency and performance are the factors on which power consumption of CPU depends. The power consumption of CPU is:

$$P_{CPU} = f_{base} + \sum_{i=0}^{n-1} (f_i * U_i) \quad (1)$$

where f_{base} and f_i are frequency-dependent coefficients, U_i is the utilization of the ith CPU and n is the number of CPUs.

3.1.2 Power Consumption of RAM

The power consumption of RAM depends on the type of modules and is calculated as:

$$P_{RAM} = P_{s1} * U + P_{s2} \quad (2)$$

where P_{s1} and P_{s2} are coefficients of power. U is the aggregated CPU utilization that is:

$$U = \sum_{j=0}^{n-1} Uj \tag{3}$$

U_j is the utilization of j^{th} CPU and n is the total number of CPUs.

3.1.3 Power Consumption of Wi-Fi

Another source of energy consumption in MDs is the power consumption of Wi-Fi. Idle, initial, send, receive and tail are states of the Wi-Fi model, with the total Wi-Fi power consumption for the MDs being the sum of their power consumption as shown in Equation 4. Coefficients are based on quad-core Galaxy S3. T_{State} is the state's time, where T_{Send} and T_{Idle} are calculated in runtime. Finally, N is the number of packets sent or received per second which considered more than 20.

$$P_{Wi\text{-}Fi} = P_{Init} + P_{Send} + P_{Receive} + P_{Tail} + P_{Idle} \tag{4}$$

where

$$P_{Init} = (0.8613 * N + 98.612) * T_{Init}$$

$$P_{Send} = (0.4049 * N + 686.93) * T_{Send}$$

$$P_{Receive} = 0.0211 * N + 15.628$$

$$P_{Tail} = 195 * T_{Tail}$$

$$P_{Idle} = 20 * T_{Idle}$$

4. The Proposed Approach

Each FD has its features and parameters based on its internal structure on which the best FD is chosen for the module placement. The parameters that were used in this paper are: authentication, confidentiality, integrity, availability, capacity, speed and cost. In this paper, we had four FDs that were called FD1, FD2, FD3 and FD4. For example, if the authentication, confidentiality, integrity, availability, capacity, speed and cost of the task that has arrived for processing are > 0.7, > 0.5, < 0.3, < 0.8, < 0.7, < 0.8 and < 0.9, respectively, then the task is performed on FD1. Checking these conditions for allocating the appropriate device can be carried out by Decision Tree or other trees. The features and values of each of the FDs are shown in Table 3.

Table 3. The features of FDs.

	Authentication	**Confidentiality**	**Integrity**	**Availability**	**Capacity**	**Speed**	**Cost**
FD1	> 0.7	> 0.5	< 0.3	< 0.8	< 0.7	< 0.8	< 0.9
FD2	<= 1	< 0.4	> 0.1	< 0.7	< 0.6	< 0.7	< 0.8
FD3	> 0.8	> 0.5	< 0.6	> 0.7	> 0.8	< 0.8	> 0.7
FD4	< 0.9	< 0.7	< 0.8	> 0.9	> 0.7	< 0.8	> 0.6
Cloud	Other values						

Table 4. Part of the dataset.

No.	**Authentication**	**Confidentiality**	**Integrity**	**Availability**	**Capacity**	**Speed**	**Cost**	**Target Classes**
1	0.1	0.16	0.32	0.94	1	0.51	0.69	FD4
2	0.91	0.61	0.04	0.18	0.26	0.17	0.53	FD1
3	0.86	0.18	0.64	0.55	0.05	0.66	0.28	Cloud
4	0.61	0.33	0.28	0.24	0.4	0.13	0.13	FD2
5	0.79	0.45	0.26	0.39	0.17	0.68	0.51	Mobile
6	0.86	0.81	0.12	0.88	0.94	0.59	0.94	FD3
…	…	…	…	…	…	…	…	…

Our base article is [41]. The simulator assigns values between 0 and 1 to features at random. Tasks are offloaded in suitable FDs and otherwise in cloud or mobile, so that when the sum of memory power consumption and CPU power consumption is less than Wi-Fi's power consumption, tasks run in mobile; otherwise, according to Table 3, they are offloaded in an FD or cloud.

We stored the outputs of the simulator as a dataset with 457 samples with 7 features and 1 target class, part of which is shown in Table 4.

In general, with more data, classifiers are better trained to classify. The target class is the device in which the task is offloaded. Hence, we had sex targets, FD1, FD2, FD3, FD4, cloud and mobile. Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers classify based on feature values and draw the plot of the tree. In the plot of the tree, each node is divided into two branches and leaves are the target classes.

For example, in this code that runs by Decision Tree, availability is in the root. If its value in the sample is greater than 0.875, it goes to the left branch; otherwise, it goes to the right branch and so on. In Random Forest, Extra-trees and AdaBoost classifiers, 100 different trees are considered by default and we choose the best one with the highest accuracy. According to the plot of them, we extracted each sequential conditions from root to leaves and insert them into the simulator in the corresponding section. This reduced the number of conditions and response time.

In practice, some of these parameters are not used. For example, the authentication parameter in FD2 is not required for classification. What these classifiers do is to improve the conditions in practice. These methods can be used for tasks that require a shorter response time. In fact, in the operational environment, the values of 7 features and 1 target class can be stored as data in a dataset. Then, by using these classifiers, we can insert the conditions effectively into the simulator for the next tasks and achieve less response time.

In Algorithm I, the steps are performed. In lines 1 to 3 FDs are created with time complexity O(k), so that k is the number of FDs and VMs created with time complexity O(t), where t is the number of VMs in line 4. Then, n tasks are taken from MDs with O(n) in line 5. The broker is created with O(1) and VMs and tasks are submitted to it in lines 6 and 7 with O(n+t). So time complexity until this stage is equal to O(k+n+t).

Algorithm I

```
Input: VMs, Tasks, FDs, cloud
Output: VMs and tasks in the broker
1: for each i ϵ FDs do
2:         Create micro DCs in $FD_i$
3: end for
4: Create VMs
5: Get tasks from MDs
6: Create broker
7: Submit VMs and tasks to the broker
```

In Algorithm II, input includes dataset according to Table 4. By executing Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers, its plot is obtained.

Algorithm II

```
Input: dataset
Output: Plot of Trees
1: Running Decision Tree  classifier by Python code to draw its plot
2: Running Random Forest, Extra-trees and AdaBoost classifiers and choosing the best one with the
heights accuracy by Python code to draw its plot
```

In Algorithm III, tasks are offloaded in one of six modes that include mobile, FD1 to FD4 and cloud. Placement is prepared by calling Algorithm I in line 1. Then, in line 2, new conditions are inserted in the corresponding section of the simulator by running Algorithm II. In lines 3 to 10, tasks are offloaded in one of the six modes with O(n), where n is the number of tasks. In Equations 1 to 4 it is shown, how to calculate memory power, CPU power and Wi-Fi’s power consumption. When the sum of memory power consumption and CPU power consumption is less than Wi-Fi’s power consumption, tasks run in mobile; otherwise, they are offloaded in an FDs or cloud.

Algorithm III

```
Input: VMs, Tasks, FDs, cloud
Output: The places of Tasks
1: Preparing placement by calling Algorithm I
2: Insert new conditions by running Algorithm II to the simulator in the corresponding section
3: for each n ϵ Tasks do
4:     Calculate $PC_{Wi\text{-}Fi}$, $PC_{CPU}$ and $PC_{RAM}$
5: if $PC_{CPU} + PC_{RAM} < PC_{Wi\text{-}Fi}$ then
6:     Execute task in MD
7: else
8:     Place task in suitable $FD_j$ (j from 1 to 4) or cloud
9: end if
10: end for
```

5. EVALUATION

In this part, we compare three methods: mobile, Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers. We used Cloudsim simulator. In the local mobile processing method, the tasks are executed in MD and don’t offload to FDs and cloud. These classifiers were executed and the plot of trees is drawn by Python code in the sci-kit learn website [7]. Then, new if statements are added to the relevant section in the simulator. Response time, power consumption of CPU, power consumption of RAM and performance were compared in these three methods. In the simulator, each of them has been run 30 times individually and the average values were presented.

5.1 Configurations of the Simulator

DC and micro DC configurations are shown in Table 5.

We executed the simulation in different states of number of VMs and tasks, as shown in Table 6. In our simulation, the main classes are Cloudlet, Datacenter, DatacenterBroker and VM. Task offloading is carried out in the DatacenterBroker class. New conditions resulting from the implementation of the Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers were inserted into Cloudlet class.

Table 5. DC and Micro DC configurations.

Name	**DC**	**Micro DC**
CPU	Octa-core	Quad-core
Memory size	8192	2048 GB
Memory cost	0.015	0.005
Storage size	1 TB	100 GB
Storage cost	0.05	0.01
Bandwidth rate	100 MB/S	10 MB/S
Bandwidth cost	0.1	0.01

Table 6. Different states of the number of VMs and tasks in Cloudsim simulator.

No.	**VMs**	**Tasks**	**No.**	**VMs**	**Tasks**
1	10	10	6	50	100
2	10	20	7	100	100
3	20	20	8	100	200
4	40	50	9	200	200
5	50	50	10	500	500

5.2 Offloading Frequency

Figure 2 shows the offloading frequency that is the frequency of offloading tasks to FDs or cloud. When the sum of memory power consumption and CPU power consumption is more than Wi-Fi’s power consumption, tasks are offloaded in FDs and cloud. According to Table 1 and the simulator assign values of seven features between 0 and 1 at random, most tasks are offloaded in cloud, FD2, FD1, FD4 and FD3, respectively.

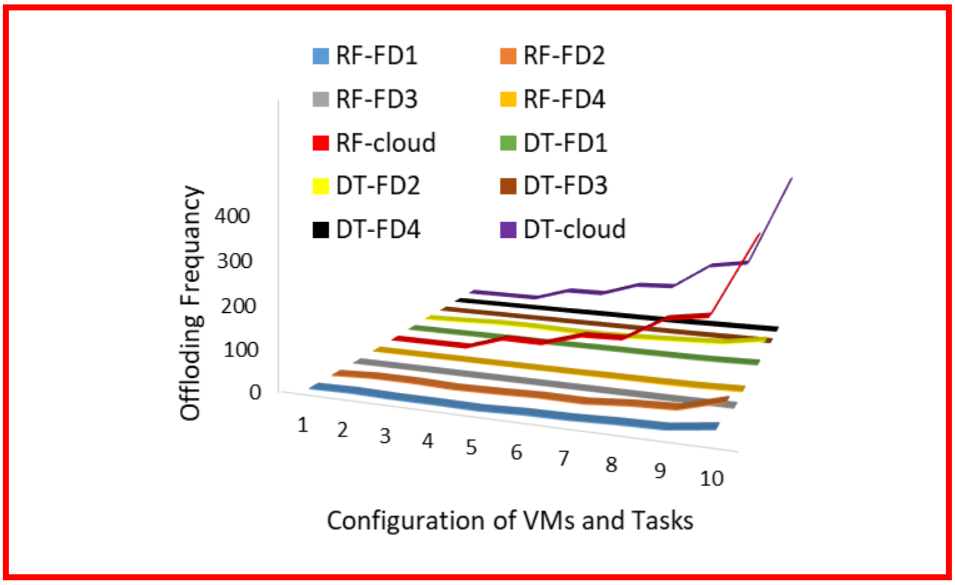


Figure 2. Comparison of offloading frequency to cloud and FDs by decision tree, random forest, extra-trees and AdaBoost and mobile methods.

5.3 Response Time

In Figure 3, the response time is shown. Response time of mobile is less than in Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers, because there is no offloading. As can be seen in the Figure, the response time of Random Forest, Extra-trees and AdaBoost methods is better than that of Decision Tree, because they consider 100 different trees by default, where we choose the best one with the highest accuracy. The average response time in Random Forest, Extra-trees and AdaBoost methods is 649.361, 646.61 and 643.452ms, respectively, while in the Decision Tree method, it is 696.363ms.

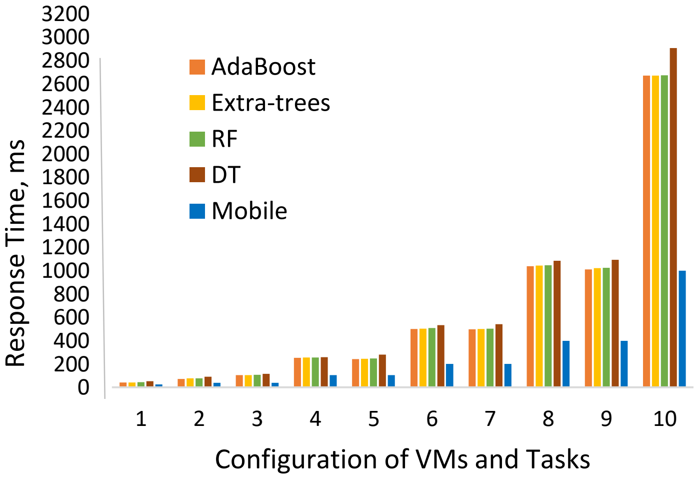


Figure 3. Comparison of response time of decision tree, random forest, extra-trees and AdaBoost and mobile methods.

5.4 The Power Consumption of CPU and RAM

Total power consumption of CPU and RAM is presented in Figures 4 and 5. On average, they are almost the same in Decision Tree, Random Forest, Extra-trees and AdaBoost methods. In the mobile method, it is higher than in the other methods. The average of power consumption of CPU in Decision Tree, Random Forest, Extra-trees and AdaBoost methods is 1874.174, 1875.074, 1877.667 and 1878.147W, respectively, while in the mobile method, it is 1898.796W. The average of the power consumption of RAM in Decision Tree, Random Forest, Extra-trees and AdaBoost methods is 112.45, 112.503, 113.034 and 113.038W, respectively, while in the mobile method, it is 113.842W.

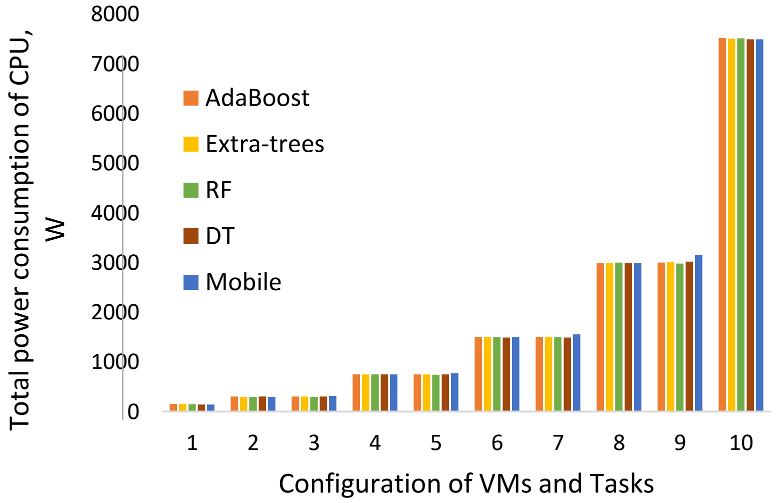


Figure 4. Total power consumption of CPU in decision tree, random forest, extra-trees and AdaBoost and mobile methods.

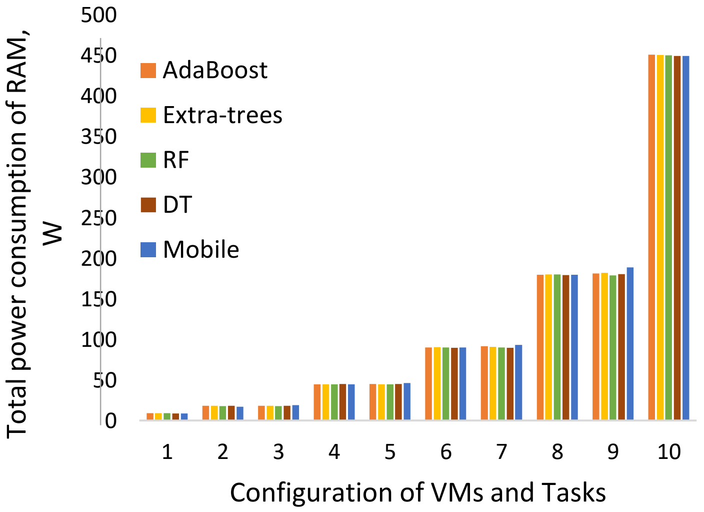


Figure 5. Total power consumption of RAM in decision tree, random forest, extra-trees and AdaBoost and mobile methods.

5.5 Performance

In Figure 6, performance is presented. Its calculation is as follows:

$$\text{performance} = 1 - (P_{Receive} + P_{Idle}) / P_{CPU} \tag{5}$$

where P_{Idle} and $P_{Receive}$ are the power consumption of Wi-Fi in the idle and receive states and P_{CPU} is the power consumption of CPU in MD (see Eq. 4) [42]. As a result, in Figure 5, the performance of Decision Tree, Random Forest, Extra-trees and AdaBoost is better than in the mobile method.

5.6 Comparison of Algorithms

Our simulation results show that the response time of the mobile method is less than in Decision Tree,

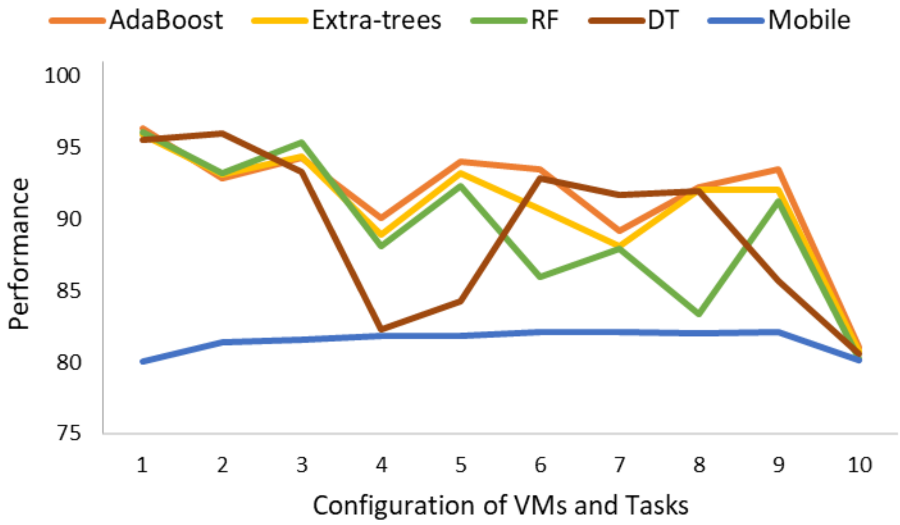


Figure 6. Performance comparison of decision tree, random forest, extra-trees and AdaBoost and mobile methods.

Random Forest, Extra-trees and AdaBoost classifiers, because there is no offloading and the response time of Random Forest, Extra-trees and AdaBoost methods is better than that of the Decision Tree method. AdaBoost is a meta-estimator that begins by fitting a classifier on the original dataset. Then, additional copies of the classifier are fitted on the same dataset. However, the weights of wrong classified instances are adjusted such that subsequent classifiers focus more on difficult cases [11]. Boosting methods train predictors consecutively and try to improve its predecessor, because they collect the predictions made by each Decision Tree within the forest. Some differences between them are in AdaBoost; the Decision Trees have a depth of 1 and the final prediction made by the model is impacted by the predictions made by each Decision Tree [12]. So, the tree of AdaBoost performs better in terms of the conditions that should be inserted in the corresponding section of the simulator compared to Decision Tree. Response time improved by 7.6 percent in this case. An Extra-trees classifier fits some randomized Decision Trees on various sub-samples of the dataset, improves the predictive accuracy and controls over-fitting implementing a meta-estimator and averaging, respectively [10]. Thus, the tree of Extra-trees performs better in terms of the conditions that should be inserted in the corresponding section of the simulator compared to Decision Tree. In this case, response time improved by 7.14 percent. Random Forest fits 100 Decision Tree classifiers by default on the dataset’s different sub-samples and improves the predictive accuracy using averaging [8]. So, the tree of Random Forest performs better in terms of the conditions that should be inserted in the corresponding section of the simulator compared to Decision Tree. Response time improved by 6.75 percent in this case.

Total power consumption of CPU and RAM is almost the same in these methods and in the mobile method, it is higher than in the other methods. Also, the performance of them is better than that of the mobile method, because there is no offloading on the mobile. Thus, our offloading methods of using Random Forest, Extra-trees and AdaBoost classifiers have a better response time than Decision Tree on MFC.

6. Conclusions

The application of computing resources through mobile devices (MDs) is called Mobile Computing. Between cloud datacentres and devices is (Mobile) Fog Computing (MFC). Tasks are offloaded in suitable FDs and otherwise in cloud or mobile. We used Decision Tree, Random Forest, Extra-trees and AdaBoost classifiers for task offloading on MFC, where Random Forest, Extra-trees and AdaBoost classifiers had a better response time than previous methods. Our simulation results showed that the response time of the mobile method is less than these classifiers, because there is no offloading and the response time of Random Forest, Extra-trees and AdaBoost methods was better than in the Decision Tree method. Thus, our offloading methods of using Random Forest, Extra-trees and AdaBoost classifiers had a better response time than that of Decision Tree on MFC. Total power consumption of CPU and RAM was almost the same in these methods and in the mobile method, it was higher than in the other methods. Also, the performance of Decision Tree, Random Forest, Extra-trees and AdaBoost was better than in the mobile method.

For future work, we will try to implement an algorithm that gives better results in the simulator. Also, machine learning methods can be a great way to task offloading.

REFERENCES

[1] T. H. Noor, S. Zeadally, A. Alfazi and Q. Z. Sheng, "Mobile Cloud Computing: Challenges and Future Research Directions," Journal of Network and Computer Applications, vol. 115, pp. 70-85, 2018.

[2] R. Mahmud, R. Kotagiri and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," Internet of Everything, pp. 103-130, Springer, Singapore, 2018.

[3] R. Roman, J. Lopez and M. Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges," Future Generation Computer Systems, vol. 78, pp. 680-698, 2018.

[4] F. Gu, J. Niu, Z. Qi and M. Atiquzzaman, "Partitioning and Offloading in Smart Mobile Devices for Mobile Cloud Computing: State-of-the-art and Future Directions," Journal of Network and Computer Applications, vol. 119, pp. 83-96, 2018.

[5] C. Li, Y. Xue, J. Wang, W. Zhang and T. Li, "Edge-oriented Computing Paradigms: A Survey on Architecture Design and System Management," ACM Computing Surveys (CSUR), vol. 51, no. 2, pp. 1-34, 2018.

[6] S. Fletcher and Md. Z. Islam, "Decision Tree Classification with Differential Privacy: A Survey," ACM Computing Surveys (CSUR), vol. 52, no. 4, pp. 1-33, 2019.

[7] Scikit-learn, "Decision Trees (DTs)," [Online], Available: https://scikit-learn.org/stable/modules/tree.html#tree-algorithms.

[8] Scikit-learn, "Random Forest Classifier," [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=random%20forest#sklearn.ensemble.RandomForestClassifier.

[9] A. B. Shaik and S. Srinivasan, "A Brief Survey on Random Forest Ensembles in Classification Model," Proc. of the International Conference on Innovative Computing and Communications, pp. 253-260, Springer, Singapore, 2019.

[10] Scikit-learn, "Extra Trees Classifier," [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[11] Scikit-learn, "AdaBoost Classifier," [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[12] Towards Data Science, "AdaBoost Classifier Example in Python," [Online], Available: https://towardsdatascience.com/machine-learning-part-17-boosting-algorithms-adaboost-in-python-d00faac6c464

[13] S. Wu, C. Mei, H. Jin and D. Wang, "Android Unikernel: Gearing Mobile Code Offloading Towards Edge Computing," Future Generation Computer Systems, vol. 86, pp. 694-703, 2018.

[14] L. Liu, Z. Chang and X. Guo, "Socially Aware Dynamic Computation Offloading Scheme for Fog Computing System with Energy Harvesting Devices," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1869-1879, 2018.

[15] L. Liu, Z. Chang, X. Guo, S. Mao and T. Ristaniemi, "Multi-objective Optimization for Computation Offloading in Fog Computing," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 283-294, 2017.

[16] Z. Tang, X. Zhou, F. Zhang, W. Jia and W. Zhao, "Migration Modeling and Learning Algorithms for Containers in Fog Computing," IEEE Trans. on Services Computing, vol. 12, no. 5, pp. 712-725, 2018.

[17] J. Du, L. Zhao, J. Feng and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-max Fairness Guarantee," IEEE Transactions on Communications, vol. 66, no. 4, pp. 1594-1608, 2018.

[18] J. Shuja, A. Gani, K. Ko, K. So, S. Mustafa, S. A. Madani and M. K. Khan, "SIMDOM: A Framework

for SIMD Instruction Translation and Offloading in Heterogeneous Mobile Architectures," Transactions on Emerging Telecommunication Technologies, vol. 29, no. 4, p. e3174, 2018.

[19] H. Yan, X. Zhang, H. Chen, Y. Zhou, W. Bao and L. T. Yang, "DEED: Dynamic Energy-efficient Data Offloading for IoT Applications under Unstable Channel Conditions," Future Generation Computer Systems, vol. 96, pp. 425-437, 2019.

[20] Md. G. R. Alam, M. M. Hassan, Md. ZIa Uddin, A. Almogren and G. Fortino, "Autonomic Computation Offloading in Mobile Edge for IoT Applications," Future Generation Computer Systems, vol. 90, pp. 149-157, 2019.

[21] M. Aazam, S. Zeadally and K. A. Harras, "Offloading in Fog Computing for IoT: Review, Enabling Technologies and Research Opportunities," Future Generation Computer Systems, vol. 87, pp. 278-289, 2018.

[22] H. Lu, C. Gu, F. Luo, W. Ding and X. Liu, "Optimization of Lightweight Task Offloading Strategy for Mobile Edge Computing Based on Deep Reinforcement Learning," Future Generation Computer Systems, vol. 102, pp. 847-861, 2020.

[23] A. Jaddoa, G. Sakellari, E. Panaousis, G. Loukas and P. G. Sarigiannidis, "Dynamic Decision Support for Resource Offloading in Heterogeneous Internet of Things Environments," Simulation Modeling Practice and Theory, vol. 101, p.102019, [Online], Available: https://doi.org/10.1016/j.simpat.2019. 102019, 2020.

[24] C. Zhang, H.-H. Cho and C.-Y. Chen, "Emergency-level-based Healthcare Information Offloading over Fog Network," Peer-to-Peer Networking and Applications, vol. 13, no. 1, pp. 16-26, 2020.

[25] K. Xiao, Z. Gao, W. Shi, X. Qiu, Y. Yang and L. Rui, "EdgeABC: An Architecture for Task Offloading and Resource Allocation in the Internet of Things," Future Generation Computer Systems, vol. 107, pp. 498-508, 2020.

[26] I. A. Elgendy, W. Zhang, Y.-C. Tian and K. Li, "Resource Allocation and Computation Offloading with Data Security for Mobile Edge Computing," Future Generation Computer Systems, vol. 100, pp. 531-541, 2019.

[27] W. Tang, X. Zhao, W. Rafique, L. Qi, W. Dou and Q. Ni, "An Offloading Method Using Decentralized P2P-enabled Mobile Edge Servers in Edge Computing," Journal of Systems Architecture, vol. 94, pp. 1-13, 2019.

[28] Q. Wang, S. Guo, J. Liu and Y. Yang, "Energy-efficient Computation Offloading and Resource Allocation for Delay-sensitive Mobile Edge Computing," Sustainable Computing: Informatics and Systems, vol. 21, pp. 154-164, 2019.

[29] X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer and S. Wan, "An Edge Computing-enabled Computation Offloading Method with Privacy Preservation for Internet of Connected Vehicles," Future Generation Computer Systems, vol. 96, pp. 89-100, 2019.

[30] M. Adhikari and H. Gianey, "Energy Efficient Offloading Strategy in Fog/Cloud Environment for IoT Applications," Internet of Things, vol. 6, p. 100053, 2019.

[31] A. Bozorgchenani, S. Disabato, D. Tarchi and M. Roveri, "An Energy Harvesting Solution for Computation Offloading in Fog Computing Networks," Computer Communications, vol. 160, pp. 577-587, 2020.

[32] Y. E. M. Hamouda, "Modified Random Bit Climbing (Λ-MRBC) for Task Mapping and Scheduling in Wireless Sensor Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 01, pp. 17-33, 2019.

[33] M. Zeng, Y. Li, K. Zhang, M. Waqas and D. Jin, "Incentive Mechanism Design for Computation Offloading in Heterogeneous Fog Computing: A Contract-based Approach," Proc. of IEEE International Conference on Communications (ICC), pp. 1-6, Kansas City, MO, USA, 2018.

[34] X. Zhu, S. Chen and G. Yang, "Energy and Delay Co-aware Computation Offloading with Deep Learning in Fog Computing Networks," Proc. of the 38th IEEE International Performance Computing and Communications Conference (IPCCC), pp. 1-6, London, UK, 2019.

[35] X. Zhao, L. Zhao and K. Liang, "An Energy Consumption Oriented Offloading Algorithm for Fog Computing," Proc. of the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, pp. 293-301, Springer, Cham, 2016.

[36] J. Shuja, S. Mustafa, R. W. Ahmad, S. A. Madani, A. Gani and M. K. Khan, "Analysis of Vector Code

Offloading Framework in Heterogeneous Cloud and Edge Architectures," IEEE Access, vol. 5, pp. 24542-24554, 2017.

[37] M. Othman, A. N. Khan, J. Shuja and S. Mustafa, "Computation Offloading Cost Estimation in Mobile Cloud Application Models," Wireless Personal Communications, vol. 97, no. 3, pp. 4897-4920, 2017.

[38] J. Shuja, A. Gani, M. Habib ur Rehman, E. Ahmed, S. A. Madani, M. K. Khan and K. Ko, "Towards Native Code Offloading Based MCC Frameworks for Multimedia Applications: A Survey," Journal of Network and Computer Applications, vol. 75, pp. 335-354, 2016.

[39] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974-983, 2014.

[40] S. Sardellitti, G. Scutari and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-edge Computing," IEEE Transactions on Signal and Information Processing over Networks, vol. 1, no. 2, pp. 89-103, 2015.

[41] D. Rahbari and M. Nickray, "Task Offloading in Mobile Fog Computing by Classification and Regression Tree," Peer-to-Peer Networking and Applications, vol. 13, pp. 104-122, 2019.

[42] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong and J. P. Jue, "All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey," Journal of Systems Architecture, vol. 98, pp. 289-330, 2019.

ملخص البحث:

إنّ تطبيق مصادر الحوسبة من خلال الأجهزة النّقّالة يُعرف بالحوسبة النّقّالة؛ وبين مراكز البيانات السّحابية والأجهزة يُسمى ذلك الحوسبة الضّبابية النّقّالة. لقد قمنا بتشغيل المحاكي () لتفريغ المهام في أجهزة حوسبة ضّبابية مناسبة أو في السّحابة أو في أجهزة نقّالة. وقمنا بتخزين مخرجات المحاكي في شكل مجموعة بياناتٍ ذات سِمات مميزة، بالإضافة الى درجة هدف. والأخيرة هي جهاز يتم فيه تفريغ المهام. أما سِمات المهام فهي: الأصالة، والسِّرية، والسَّلامة (الكمال)، والتّوافر، والسِّعة، والتكلفة.

تم استخدام عددٍ من المصنفات (AdaBoost; Extra-trees; RF; DT) ومقارنتها بناءً على قيم السِّمات، الى جانب رسم مخططات الشجرة لكل منها. ووفقاً للرسومات التي تم الحصول عليها من تلك المصنفات، تم استخلاص الوضع التتابعي لكلٍّ منها من "الجذر" وحتى "الأوراق" وإدخالها الى المحاكي. والجدير بالذكر أن ما تقوم به هذه المصنفات هو تحسين الشروط التي يتعين إدخالها الى الجزء المناظر من المحاكي. وقادت النتائج الى تحسين زمن الاستجابة للتفريغ باستخدام مصنفات RF و Extra-trees و AdaBoost على نحوٍ يفوق التحسين المتحقق باستخدام مصنف DT.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

CHANNEL ESTIMATION AND DETECTION FOR OFDM MASSIVE-MIMO IN FLAT AND FREQUENCY-SELECTIVE FADING CHANNELS[1]

Abdelhamid Riadi[1], Mohamed Boulouird[2] and Moha M'Rabet Hassani[1]

(Received: 2-May-2020, Revised: 2-Jul.-2020 and 2-Aug.-2020, Accepted: 28-Aug.-2020)

ABSTRACT

In this paper, the least-squares channel estimation (LSCE) is investigated for Massive-Multiple-Input Multiple-Output (Ma-MIMO) OFDM systems based on pilot tones. The uplink (UL) transmission is considered, in which a channel estimation approach is proposed by forming a matrix equation with all the unknown channel parameters (UCPs) in one vector and estimating that vector by the least-square (LS). The mean square error (MSE) of the LSCE is computed. Flat fading and frequency-selective fading are evaluated for single and multiple OFDM symbols, concerning this MSE. The requirement of the pilot sequence is investigated in flat fading and frequency-selective fading. Besides, it is shown that the number of pilots exhibits desirable trade-offs between the base station (BS) antenna and channel taps. Rayleigh and Rician channel fading is considered to evaluate the system performance with different channel taps. Performances are compared in terms of Bit Error Rate (BER). Moreover, to enhance linear detector performance, nonlinear detectors are used. The requirement of pilot sequence and the increased receive diversity provide a lower BER for the nonlinear detector.

KEYWORDS

Massive MIMO, OFDM, Flat and frequency selective fading, OSIC, Rician channel, Rayleigh channel.

1. INTRODUCTION

Wireless mobile networks are classified into three broad categories. The first is satellite technologies (DVB-S2, TS2...), the second is wireless technologies (PAN, WLAN, WMAN, Wimax, ...) and the third is cellular technologies (GSM, GPRS, UMTS, LTE, 5G, ...). 5G is a promising technology based on Ma-MIMO between transceiver. Recently, in a world of great mobility, the speed and capacity of communication systems are essential elements to keep people from all over the world in communication. Ma-MIMO is also known as very large MIMO, ARGOS, large-scale antenna systems, full-dimension MIMO and hyper MIMO [1]. It's a promising emerging communication technology for 5G cellular networks [2]. Increasing the number of antennas at the BS, Ma-MIMO combined with orthogonal frequency-division multiplexing (OFDM) can support very high throughput and/or performance of the links as well as spectral efficiency [3]. In an OFDM system, like all wireless communication systems, the received signal is usually distorted by the channel characteristics. To recover the transmitted bits, the channel effect must be estimated and compensated in the receiver [4]. In fact, in a real environment, several multi-path phenomena degrade system performances. Hence, these phenomena include both small-scale fading and large-scale fading [1], [3], [5] and [6]. Channel estimation is a technique referred to in order to know channel properties. It is a very important technique in multi-cell multi-user Ma-MIMO systems. A better improvement of spectral and energy efficiency is achieved using time division duplex with pilot contamination in Rayleigh fading channels [7], while the LSCE technique is widely used for channel estimation. In addition to that, precoding at the transmitter side has a greater effect to facilitate the use of the linear detector and can help decrease complexity at the receiver [8].

On the other hand, the analytical BER performance for BPSK has been discussed for the Zero-Forcing (ZF) detector. It's associated with the Successive Interference Cancellation (SIC) for an arbitrary

[1] This paper is an extended version of a conference paper [27] "ZF/MMSE and OSIC Detectors for Uplink OFDM Massive MIMO Systems," Proc. of IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 2019.

1. A. Riadi and M. M. Hassani are with Instrumentation, Signals and Physical Systems (I2SP), Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco, E-mails: abdelhamid.riadi@edu.uca.ac.ma and hassani@ucam.ac.ma
2. M. Boulouird is with National School of Applied Sciences of Marrakech (ENSA-M), Cadi Ayyad University, Marrakech, Morocco, E-mail: m.boulouird@uca.ac.ma

number of transmitting and receiving antennas [9]. Similarly, the high-order M-QAM mode with ZF-OSIC receiver is discussed in [10]. Reduced complexity for MIMO Receiver and combined with the ZF-OSIC is discussed in [11], as well as in the literature, before applying the matrix inversion. The residual interference cancellation error covariance matrix of the off-diagonal elements is ignored first and the reduced-complexity approximations of soft-output MMSE-OSIC MIMO detector are studied in [12]. In addition to that, a novel soft-output MIMO MMSE OSIC detector under channel estimation is proposed in [13]. Further, a low-complexity MMSE-OSIC detector called MMSE-OBEP (Ordering Based on Error Probability) to decrease the BER of MMSE-OSIC detectors is evaluated in [14].

The performance of wireless communication systems is mainly governed by the wireless channel environment. Due to the constructive and destructive interference of multiple signal paths (multi-paths), the wireless channel is rather unpredictable. In Ma-MIMO system, the channel can be estimated by using a preamble or pilot symbol known at the receiver. Then, various interpolation techniques are employed to estimate the channel response of the subcarriers between pilot tones.

In this work, our contributions are summarized as follows:

- We propose a channel estimation approach by forming a matrix equation with all the unknown channel parameters in one vector.
- This vector is estimated in UL transmission using least-square (LS) method.
- Training symbols can be used for channel estimation.
- Constraints on pilot sequences for various scenarios are derived with respect to MSE.

This paper is organized as follows. In Section 2, a system model in the UL transmission is illustrated, in which a single cell with a BS of several antennas is considered. Section 3 is devoted to the LSCE for a Ma-MIMO system, while in Section 4, the MSE of the LSCE is described. In the same way, flat fading and frequency-selective fading according to pilot sequence requirements are investigated. Section 5 presents linear detectors. In Section 6, to improve linear detector performance, OSIC detectors are applied. Section 7 presents the simulation and results. Finally, we conclude this paper in Section 8.

2. COMMUNICATION SCHEME

Ma-MIMO system is considered in UL transmission from N_t users with a single antenna to a single BS with N_r antennas. The system is presented in Figure 1, for a Ma-MIMO OFDM with K sub-carriers. A cyclic prefix (CP) with length ν is inserted to form a complete OFDM symbol. The CP is considered to be larger than the largest multi-path delay [18]-[19], [27]-[30] and [33].

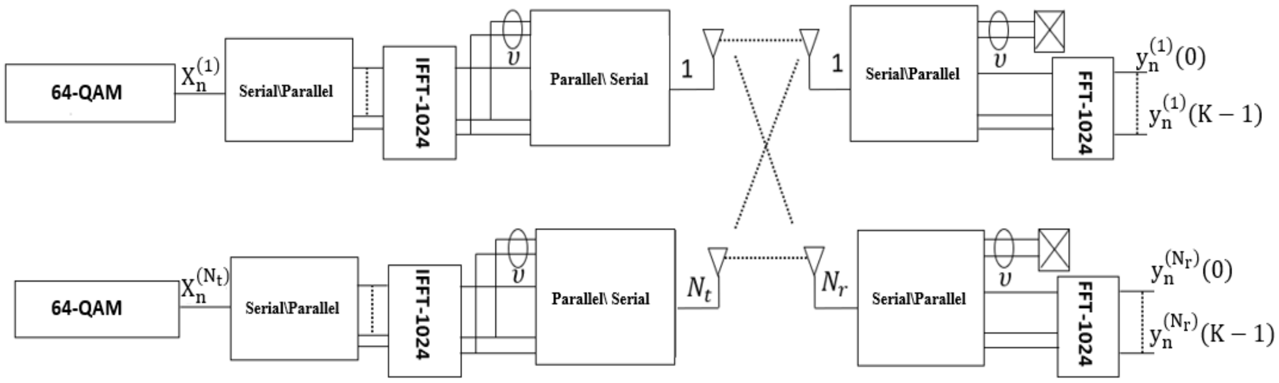


Figure 1. System model of Ma-MIMO combined with OFDM technique.

From Figure 1, the frequency response at the k^{th} subcarrier corresponding to the channel from the m^{th} transmit antenna to the q^{th} receive antenna at the n^{th} time frame (i.e., OFDM symbol) is given by [33]:

$$\begin{aligned} H_n^{(q,m)}(k) &= \sum_{l=0}^{L-1} h_n^{(q,m)}(l) e^{-j\frac{2\pi kl}{K}} \\ &= \sum_{l=0}^{L-1} h_n^{(q,m)}(l)\ W_K^{(k)(l)} \end{aligned} \tag{1}$$

where $h_n^{(q,m)}(l)$ is the l^{th} channel impulse of $h_n^{(q,m)} = \left[h_{\mathrm{n}}^{(q,m)}(0), \cdots, h_n^{(q,m)}(L-1)\,\right]^T$ $with$ dimension $(L \times 1)$ [35], which is the channel impulse response from the m^{th} transmit antenna to the q^{th} receive antenna, when the n^{th} OFDM symbol is transmitted. Under the assumption that $h_n^{(q,m)}(l)$ follows the $\mathcal{CN}(0, \sigma^2)$ [30], it can be represented by Rayleigh or Rician distribution depending on the topography of the environment.

2.1 Rayleigh and Rician Channel Fading

Rayleigh and Rician models are two channel models widely used in wireless communications. The Rician channel assumes that the transmission paths from the transmitter to the receiver are comprised of the dominant line of sight (LoS) path and other scattering paths. However, the Rayleigh channel consists of scattering channels from the transmitter to the receiver [17].

From Equation (1), the fading process $h_n^{(q,m)}(l) = |h_n^{(q,m)}(l)|e^{-j\phi_n(l)}$ is assumed to be a zero-mean complex Gaussian process [36], with uniformly distributed phase $\phi_n(l)$ on $[0, 2\pi]$ and with Rayleigh distributed envelope $|h_n^{(q,m)}(l)|$; whereas the magnitude $|h_n^{(q,m)}(l)|$ of the l^{th} tap is a Rayleigh random variable with the probability density function (pdf) [17] and [34]:

$$pdf(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \qquad x \geq 0 \tag{2}$$

and the squared magnitude $\left|h_n^{(q,m)}(l)\right|^2$ is exponentially distributed with density:

$$\frac{x}{\sigma^2} e^{-\frac{\mathrm{x}}{\sigma^2}}, \qquad x \geq 0 \tag{3}$$

This model, which is called Rayleigh fading, is quite reasonable for scattering mechanisms, where there are many small reflectors, but it is adopted primarily for its simplicity in typical cellular situations with a relatively small number of reflectors. The word Rayleigh is almost universally used for this model, but the assumption is that the tap gains are circularly symmetric complex Gaussian random variables [17].

There is a frequently used alternative model, in which the LoS path (often called a specular path) is large and has a known magnitude and there are also a large number of independent paths. In this case, $h_n^{(q,m)}(l)$, at least for one value of l, can be modeled as [17]:

$$h_n^{(q,m)}(l) = \sqrt{\frac{K}{K+1}}\sigma e^{j\theta} \;+ \sqrt{\frac{1}{K+1}}\mathcal{CN}(0, \sigma^2) \tag{4}$$

with the first term corresponding to the specular path arriving with uniform phase θ and the second term corresponding to the aggregation of the large number of reflected and scattered paths, independent of θ. The parameter K (so-called K-factor) is the ratio of the energy in the specular path to the energy in the scattered paths; the larger K is, the more deterministic is the channel. The magnitude of such a random variable is said to have a Rician distribution. Its pdf is defined by [17]:

$$pdf(x) = \frac{x}{\sigma^2} e^{-\frac{x^2+c^2}{2\sigma^2}} I_0\left(\frac{xc}{\sigma^2}\right) \tag{5}$$

where c represents the LoS component and $I_0(.)$ is the modified zero-order Bessel function of the first kind. From Equation (5), the Rician factor is defined as $= \frac{c^2}{2\sigma^2}$. Moreover, where K = 0, that is to say no line of sight (NLoS) environment, Equation (5) reduces to Equation (2) of Rayleigh pdf [17]; whereas, in the next, the q^{th} receive antenna is considered at n^{th} time frame.

Based on Figure (1) and Equation (1), Ma-MIMO model is defined in the second subsection.

2.2 Massive MIMO Model

After removing the CP at the q^{th} receive antenna (Figure 1) and using Equation (1), the received signal $y_{\mathrm{n}}^{(q)}(k)$ can be written as [19], [27]-[30]:

$$y_{\mathrm{n}}^{(q)}(k) = \sum_{m=1}^{N_t} H_n^{(q,m)}(\mathrm{k}) x_n^{(m)}(k) + z_{\mathrm{n}}^{(q)}(k) \tag{6}$$

where $\mathrm{q} = \{1, \cdots, \mathrm{N_r}\}$, $\mathrm{k} = \{0, \cdots, \mathrm{K}-1\}$ and $n \in \{0, \cdots, \mathrm{g}-1\}$. As is clear from Equation (6) and a K-subcarrier OFDM, the received signal can be rewritten as:

$$\mathrm{Y}_{n}^{(\mathrm{q})} = \sum_{m=1}^{N_t} X_{n}^{(m)} \mathrm{G}_{\mathrm{n}}^{(\mathrm{m})} + \mathrm{Z}_{n}^{(\mathrm{q})} \tag{7}$$

where $\mathrm{Y}_{n}^{(\mathrm{q})} = \left[y_{\mathrm{n}}^{(q)}(0), y_{\mathrm{n}}^{(q)}(1), \cdots, y_{\mathrm{n}}^{(q)}(K-1) \right]^T$, $\mathrm{Z}_{n}^{(\mathrm{q})} = \left[z_{\mathrm{n}}^{(q)}(0), z_{\mathrm{n}}^{(q)}(1), \cdots, z_{\mathrm{n}}^{(q)}(K-1) \right]^T$.

Furthermore, The OFDM symbol that is transmitted from the m^{th} antenna at n^{th} time frame is defined by:

$$X_{\mathrm{n}}^{(m)} = \begin{bmatrix} x_{\mathrm{n}}^{(m)}(0) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & x_{n}^{(m)}(K-1) \end{bmatrix}_{K \times K}$$

where the k^{th} diagonal element of $X_{n}^{(m)}$ is $x_{n}^{(m)}(k)$. In addition to that, $\mathrm{G}_{\mathrm{n}}^{(\mathrm{m})}$ is a vector equal to $\mathrm{F}h_{n}^{(q,m)}$ of dimension $(K \times 1)$. From Equation (1), we can define:

$$G_{n}^{(m)} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & W_K^1 & \dots & W_K^{(L-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_K^{(K-1)} & \cdots & W_K^{(K-1)(L-1)} \end{bmatrix}_{K \times L} \quad h_{\mathrm{n}}^{(q,m)} = \mathrm{F} \begin{bmatrix} h_{\mathrm{n}}^{(q,m)}(0) \\ \vdots \\ h_{n}^{(q,m)}(L-1) \end{bmatrix}_{L \times 1} \tag{8}$$

Inserting Equation (8) into Equation (7), the following expression of the received signal is obtained:

$$\mathrm{Y}_{\mathrm{n}}^{(\mathrm{q})} = \sum_{m=1}^{N_t} X_{n}^{(m)} \mathrm{F} h_{\mathrm{n}}^{(q,m)} + \mathrm{Z}_{\mathrm{n}}^{(q)} \tag{9}$$

where $Z_{n}^{(q)}$ is an AWGN with zero mean and variance of σ_n^2. Based on Equation (9), the LSCE will be derived in the next section.

3. Least Square Channel Estimation

In this section, given is a K-subcarrier OFDM with a superimposed pilot sequence $\mathrm{B}_{n}^{(\mathrm{m})}$ and data sequence $\mathrm{D}_{n}^{(\mathrm{m})}$. Let us denote $X_{n}^{(m)} = \mathrm{D}_{\mathrm{n}}^{(\mathrm{m})} + \mathrm{B}_{n}^{(\mathrm{m})}$. Thus, Equation (9) can be obtained such as [27]-[30]:

$$\begin{aligned} Y_{n}^{(q)} &= \textstyle\sum_{m=1}^{N_t} X_{\mathrm{n}}^{(m)} F h_{n}^{(q,m)} + \mathrm{Z}_{\mathrm{n}}^{(q)} \\ &= \textstyle\sum_{m=1}^{N_t} (B_{\mathrm{n}}^{(m)} + D_{\mathrm{n}}^{(m)}) F h_{n}^{(q,m)} + \mathrm{Z}_{\mathrm{n}}^{(q)} \\ &= \mathcal{A}_n \mathfrak{h}_{n}^{(q)} + \mathcal{T}_n \mathfrak{h}_{\mathrm{n}}^{(q)} + \mathrm{Z}^{(q)}(n) \end{aligned} \tag{10}$$

where $\mathcal{A}_n = [B_{\mathrm{n}}^{(1)}\mathrm{F}, \cdots, B_{n}^{(N_t)}\mathrm{F}]$ of dimension $(K \times LN_t)$, $\mathcal{T}_{\mathrm{n}} = \left[D_{n}^{(1)}\mathrm{F}, \cdots, D_{\mathrm{n}}^{(N_t)}\mathrm{F} \right]$ of dimension $(K \times LN_t)$. Similarly, $B_{n}^{(\mathrm{m})}$ and $D_{n}^{(\mathrm{m})}$ are a $(K \times K)$ diagonal matrix defined as:

$$B_{\mathrm{n}}^{(m)} = \begin{bmatrix} b_{\mathrm{n}}^{(m)}(0) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & b_{n}^{(m)}(K-1) \end{bmatrix}_{K \times K} \quad \text{and} \quad D_{\mathrm{n}}^{(m)} = \begin{bmatrix} d_{\mathrm{n}}^{(m)}(0) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_{n}^{(m)}(K-1) \end{bmatrix}_{K \times K} \tag{11}$$

where $b_{n}^{(m)}(k)$ and $d_{n}^{(m)}(k)$ are the k^{th} diagonal element of $B_{n}^{(m)}$ and $D_{\mathrm{n}}^{(m)}$, respectively. Furthermore, the channel vector from all N_t users to q^{th} receive antenna can be noted as:

$$\hbar_{\mathrm{n}}^{(q)} = [h_{n}^{(q,1)^H}, \cdots, h_{n}^{(q,N_t)^H}]^H \tag{12}$$

where Equation (12) is unknown channel of dimension $(\mathrm{LN_t} \times 1)$. After forming a matrix equation with all the UCPs in one vector, the LSCE technique is applied to estimate this vector [1], [8] and [27]-[30]. Hence, the multiple channels can be estimated by:

$$\hat{\mathcal{h}}_{\mathrm{n}}^{(\mathrm{q})} = \mathcal{A}_n^{+} Y_n^{(q)} \tag{13}$$

where $\mathcal{A}_{\mathrm{n}}^{+} = (\mathcal{A}_{\mathrm{n}}^{H}\mathcal{A}_{\mathrm{n}})^{-1}\mathcal{A}_n^{H}$ is the pseudo-inverse with a full column rank of LN_t (i.e., $rank(\mathcal{A}_{\mathrm{n}}) = min(gK, LN_t)$). Otherwise, the LS method is widely used. Low-complexity and no priori statistical knowledge about the channel and the noise are required [1], [8], [17]-[19] and [30]. Hence, using Equation (10), the estimated channels can be written as:

$$\hat{\mathcal{h}}_{\mathrm{n}}^{(\mathrm{q})} = \mathcal{h}_n^{(q)} + \mathcal{A}_n^{+}\mathcal{T}_n\mathcal{h}_n^{(q)} + \mathcal{A}_n^{+}\mathrm{Z}_n^{(q)} \tag{14}$$

Further, to suppress the interference due to the data, the following condition is imposed [30]:

$$\mathcal{A}_n^{+}\mathcal{T}_n = 0_{LN_t \times LN_t} \tag{15}$$

This condition is valid when $\mathrm{B}_n^{(\mathrm{m})^{\mathrm{H}}} D_n^{(s)} = 0_{k\times K}, \forall\, m, s \in \{1, \cdots, N_t\}$ and $\forall\, n \in \{0, \cdots, \mathrm{g}-1\}$. Furthermore, to satisfy this condition, disjoint sets of pilot tones are chosen for training and data in each OFDM symbol, (i.e., zeros in $B_n^{(m)}$, where $D_{\mathrm{n}}^{(m)}$ contains non-zeros and inversely) [30]. In addition to that, let us assume P/g pilot per OFDM symbol. Thus, the dimension of $\mathcal{A}_n^{+}$ becomes $(\mathrm{P} \times \mathrm{LN_t})$ and the diagonal matrix $\mathrm{B}_n^{(m)}$ of dimension $(\mathrm{P/g} \times \mathrm{P/g})$ is at the $\mathrm{n^{th}}$ time frame. Consequently, we can write Equation (14) as:

$$\hat{\mathcal{h}}_{\mathrm{n}}^{(\mathrm{q})} = \mathcal{h}_n^{(q)} + \mathcal{A}_n^{+}\mathrm{Z}_n^{(q)} \tag{16}$$

Equation (16) indicates that $\hat{\mathcal{h}}_n^{(\mathrm{q})}$ is a combination of the true channel vector $\mathcal{h}_n^{(q)}$ plus a term affected only by the noise in the system. For zero-mean noise, $E\{\hat{\mathcal{h}}_{\mathrm{n}}^{(\mathrm{q})}\} = \mathcal{h}_{\mathrm{n}}^{(q)} + \mathcal{A}_n^{+}E\{\mathrm{Z}_{\mathrm{n}}^{(q)}\} = \mathcal{h}_n^{(q)}$; i.e., $\hat{\mathcal{h}}_n^{(\mathrm{q})}$ forms an unbiased estimate of $\mathcal{h}_n^{(q)}$. Furthermore, the estimated channel matrix $\widehat{\mathbb{H}}_n \in \mathbb{C}^{N_r\times N_t}$, including all users antennas N_t and all BS antennas N_r, is given by [30]:

$$\widehat{\mathbb{H}}_{\mathrm{n}} = \begin{bmatrix} \hat{h}_n^{(1,1)} & \cdots & \hat{h}_n^{(1,N_t)} \\ \vdots & & \vdots \\ \hat{h}_n^{(q,1)} & \cdots & \hat{h}_n^{(q,N_t)} \\ \vdots & & \vdots \\ \hat{h}_n^{(N_r,1)} & \cdots & \hat{h}_n^{(N_r,N_t)} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{h}}_{\mathrm{n}}^{(1)^T} \\ \vdots \\ \hat{\mathcal{h}}_{\mathrm{n}}^{(N_r)^T} \end{bmatrix} = [\widehat{\mathcal{H}}_n^{(1)}, \cdots, \widehat{\mathcal{H}}_n^{(N_t)}] \tag{17}$$

where the estimated channel vector at the $\mathrm{n^{th}}$ time frame and user position i is given by $\widehat{\mathcal{H}}_n^{\mathrm{i}} = [\hat{h}_n^{(1,\mathrm{i})^T}, \cdots, \hat{\mathrm{h}}_n^{(N_r,i)^T}]^T$; whereas training OFDM symbols are used over the time indices n∈{0,⋯,g-1} [1], [8] and [27]-[30]. Thus, Equation (10) can be written as: $Y^{(q)} = \mathcal{A}\,\mathcal{h}^{(q)} + \mathcal{T}\,\mathcal{h}^{(q)} + \mathrm{Z}^{(q)}$, where the received signal at the q^{th} receive antenna can be noted by: $Y^{(q)} = [Y_0^{(q)^T}, \cdots, Y_{\mathrm{g}-1}^{(q)^T}]^T$ and the noise vector becomes $\mathrm{Z}^{(q)} = [\mathrm{Z}_0^{(q)^T}, \cdots, \mathrm{Z}_{\mathrm{g}-1}^{(q)^T}]^T$. The channel vector is $\mathcal{h}^{(q)} = [\mathcal{h}_0^{(q)}, \cdots, \mathcal{h}_{\mathrm{g}-1}^{(q)}]$. Similarly, $\mathcal{A}$ and $\mathcal{T}$ are noted as:

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_0 \\ \vdots \\ \mathcal{A}_{\mathrm{g}-1} \end{bmatrix} = \begin{bmatrix} B_0^{(1)}F & \cdots & B_0^{(N_t)}F \\ \vdots & & \vdots \\ B_{\mathrm{g}-1}^{(1)}F & \cdots & B_{\mathrm{g}-1}^{(N_t)}F \end{bmatrix} \text{ and } \mathcal{T} = \begin{bmatrix} \mathcal{T}_0 \\ \vdots \\ \mathcal{T}_{\mathrm{g}-1} \end{bmatrix} = \begin{bmatrix} D_0^{(1)}F & \cdots & D_0^{(N_t)}F \\ \vdots & & \vdots \\ D_{\mathrm{g}-1}^{(1)}F & \cdots & D_{\mathrm{g}-1}^{(N_t)}F \end{bmatrix}$$

respectively. The same process of channel estimation has been carried out for all n as the $\mathrm{n^{th}}$ time frame. Moreover, after channel estimation at the q^{th}receive antenna and n^{th} time frame (i.e., Equation 13), the MSE is derived in the next section.

4. MEAN SQUARE ERROR OF LS ESTIMATOR

In this section, the MSE of LSCE is computed. Hence, the MSE of LSCE is given as [18], [21]:

$$\begin{aligned} MSE_n &= \frac{1}{LN_t} E\left\{\left\|\widehat{\mathcal{h}}_n^{(\mathrm{q})} - \mathcal{h}_n^{(q)}\right\|^2\right\} \\ &= \frac{1}{LN_t} E\left\{\left\|\mathcal{A}_n^{+} \mathrm{Z}_n^{(q)}\right\|^2\right\} \\ &= \tfrac{1}{LN_t} tr\{\mathcal{A}_n^{+} E\left(\mathrm{Z}_n^{(q)} \mathrm{Z}_n^{(q)^H}\right) \mathcal{A}_n^{+^H}\} \end{aligned} \quad (18)$$

For zero-mean white noise, we have: $E\left(\mathrm{Z}_n^{(q)} \mathrm{Z}_n^{(q)^H}\right) = \sigma_n^2$. Then, the MSE can be written as:

$$MSE_n = \tfrac{\sigma_n^2}{LN_t} tr\{(\mathcal{A}_\mathrm{n}^H \mathcal{A}_\mathrm{n})^{-1}\} \quad (19)$$

Using a similar argument as in [18], [21] and [27]-[29], we can show that in order to obtain the minimum MSE of the LSCE subject to a fixed power $\mathcal{P}$ dedicated for training, we require $\mathcal{A}_\mathrm{n}^H \mathcal{A}_\mathrm{n} = \mathcal{P} I_{LN_t \times LN_t}$. The minimum MSE is given by:

$$MSE_n^{min} = \tfrac{\sigma_n^2}{\mathcal{P}} \quad (20)$$

In the next part of this paper, flat fading and frequency-selective fading are investigated with respect to this MSE.

4.1 Flat Fading and Frequency-selective Fading

In this subsection, flat fading and frequency-selective fading are presented, with regard to MSE of the LSCE. At first, g = 1 (i.e., one OFDM symbol) is considered and then, the study is extended to multiple OFDM symbols (i.e., g > 1). When g = 1, training over the time index n = 0 and $\mathcal{A}_\mathrm{n}^H \mathcal{A}_n$ can be rewritten as[18], [21]:

$$\mathcal{A}_\mathrm{n}^H \mathcal{A}_\mathrm{n} = \begin{bmatrix} C_{1,1} & \cdots & C_{1,N_t} \\ \vdots & \ddots & \vdots \\ C_{N_t,1} & \cdots & C_{N_t,N_t} \end{bmatrix} \quad (21)$$

where $C_{m,s}$ is a sub-matrix of $\mathcal{A}_\mathrm{n}^H \mathcal{A}_\mathrm{n}$, with dimension (L×L), that is given by:

$$C_{m,s} = F^H B^{(m)}{}_0^H B_0^{(s)} F \quad (22)$$

As previously mentioned, in order to obtain the minimum MSE of the LSCE, it is necessary to fix the power $\mathcal{P}$ for training. Hence, $\mathcal{A}_\mathrm{n}^H \mathcal{A}_\mathrm{n} = \mathcal{P} I_{LN_t \times LN_t}$; that is to say [18], [21]:

$$\mathcal{A}_\mathrm{n}^H \mathcal{A}_\mathrm{n} = \begin{cases} \mathcal{P} I_{L\times L}, & if\ m = s \\ 0_{L\times L}, & if\ m \neq s \end{cases} \quad (23)$$

The positions of the P pilot tones used for training are defined as $\{k_0, k_1, k_2, \cdots, k_{P-1}\}$. In addition to that, F can be noted as F = $[f_0, f_1, f_2, \cdots, f_{L-1}]$, where $f_l = \left[e^{-\frac{j2\pi l k_0}{K}}, e^{-\frac{j2\pi l k_1}{K}}, e^{-\frac{j2\pi l k_2}{K}}, \cdots, e^{-\frac{j2\pi l k_{P-1}}{K}}\right]^T$. Thereby, from Equation (23), when m = s, the power on the i^{th} pilot tone of the m^{th} transmit antenna can be noted as p_i^m. Thus, $\sum_{i=0}^{P-1} p_i^m = \mathcal{P}$ [18], [19] and [21]. Therefore, Equation (22) can be written as follows:

$$C_{m,m} = F^H diag\{[p_0^m, p_1^m, \cdots, p_{P-1}^m]^H\} F \quad (24)$$

From Equation (24), the $(n,d)^{th}$ entry of the sub-matrix $C_{m,m}$ is obtained as:

$$\left[C_{m,m}\right]_{n,d} = f_n^H diag\{[p_0^m, p_1^m, \cdots, p_{P-1}^m]^H\} f_d \quad (25)$$

which is equivalent to:

$$\left[C_{m,m}\right]_{n,d} = \begin{cases} \mathcal{P}, & if\ n = d \\ \sum_{i=0}^{P-1} p_i^m e^{-\frac{j2\pi k_i (n-d)}{K}}, & if\ n \neq d \end{cases} \quad 26)$$

Thereby, to satisfy the first part of Equation (23), we need:

$$\sum_{i=0}^{P-1} p_i^m e^{-\frac{2j\pi k_i \phi}{K}} = \mathcal{P}\delta(\phi), \ \ \forall \phi \in \{-L+1, \cdots, L-1\} \quad (27)$$

Accordingly, the above condition is satisfied if and only if the following conditions are satisfied [18], [19] and [21]:

- $p_i^m = \frac{\mathcal{P}}{P}, \forall\, i \in \{0,1,\cdots,P-1\}\ and\ \forall\, m \in \{1,\cdots,N_t\}$
- $k_i = p_0 + pV, \forall\phi \in \{-L+1,\cdots,L-1\}\backslash\{0\},\ where\ V \in\ \mathbb{Z}\ such\ that\ PV\phi/K \in \mathbb{Z}\ and\ V\phi/K \notin\ \mathbb{Z}, \forall\phi \in \{-L+1,\cdots,L-1\}, and\ p_0 \in \{0,1,\cdots,V-1\}\ is\ some\ offset.$

Table 1. Pilot sequence for various scenarios.

Configurations	Pilot sequence requirement
Flat Fading: L=1	Equipowered+ Equispaced+ Orthogonal
Frequency-selective Fading: L>1	Equipowered+ Equispaced+ Phase Shift Orthogonal $\forall\phi \in \{-L+1,\cdots,L-1\}$

Hence, the first condition means that the pilot tone must be equipowered. Moreover, the second condition means that the pilot tones must be equispaced. For a minimum number of pilot tones or a maximum spacing, we have PV = K or V = K/P. For a practical system with inexpensive, fast and simple implementation of the FFT, the number of subcarriers must be power of 2, on the one hand. On the other hand, the P pilot tones should divide K (i.e., P must be power of 2). Hence, $P \geq LN_t$; that is to say $P = 2^{\log_2 LN_t}$ [19]. In the case where L= 1 (i.e., flat fading) pilot sequences must be equipowered, equispaced and orthogonal for various transmit antennas. When L > 1 (i.e., frequency-selective fading), the pilot sequences must be equipowered, equispaced and phase shift orthogonal for various transmit antennas (Table 1) [18], [21].

When, g > 1 (i.e., multiple OFDM symbols) is considered and training over the time indices $n \in \{0,\cdots,\mathrm{g}-1\}$, Equations (22) and (27) become:

$$C_{m,s} = \sum_{n=0}^{g-1} F_n^H {B^{(m)}}_n^H B_n^{(s)} F_n \tag{28}$$

$$\sum_{n=0}^{g-1} \sum_{i=0}^{P-1} p_{i,n}^m e^{-\frac{2j\pi k_{i,n}\phi}{K}} = \mathcal{P}\delta(\phi),\ \ \forall\phi \in \{-L+1,\cdots,L-1\} \tag{29}$$

where the positions of the P pilot tones used for training become $\{k_{0,n}, k_{1,n}, k_{2,n},\cdots,k_{P-1,n}\}$ and $F_n = [f_{0,n}, f_{1,n}, f_{2,n},\cdots,f_{L-1,n}]$, where $f_{l,n} = \left[e^{-\frac{j2\pi lk_{0,n}}{K}},\ e^{-\frac{j2\pi lk_{1,n}}{K}},\ e^{-\frac{j2\pi lk_{2,n}}{K}},\cdots,e^{-\frac{j2\pi lk_{P-1,n}}{K}}\right]^T$. The power on the i^{th} pilot tone of the m^{th} transmit antenna can be rewritten as: $p_{i,n}^m$, thus $\sum_{n=0}^{g-1}\sum_{i=0}^{P-1} p_{i,n}^m = \mathcal{P}$ [18], [19] and [21]. Furthermore, after getting the channel estimation at the BS, the requirement of the pilot sequence in flat fading and frequency-selective fading id defined. The data is detected using linear and nonlinear detectors. The detection method treats all transmitted signals as interferences, except for the desired stream from the target transmit antenna. Therefore, interference signals from other transmit antennas are minimized or nullified in the course of detecting the desired signal from the target transmit antenna, given the knowledge of the received vector and the channel matrix (i.e., Equation 17). To facilitate the detection of the desired signal from each antenna, the effect of the channel is inverted by a transformation matrix T.

5. Linear Detectors

In UL transmission, the data sequence is generated by linear detectors after channel estimation is carried out at the BS in order to obtain data sequences. Linear methods are used to generate data sequences of transmitted symbols through a linear transformation of the received vector y [22], [27]-[30]. These methods take the form of d = Ty, where T is a transformation matrix, as shown in Figure 2.

5.1 Zero Forcing Detector

The Zero Forcing (ZF) detector is a simple linear detector, in which the linear transformation on the received vector is carried out using the pseudo-inverse of the $\widehat{\mathbb{H}}_n$ matrix (i.e., Equation 17). The ZF detector completely cancels the interference from other signals (hence the name zero-forcing or interference-nulling detector) [2], [22]-[23] and [27]-[29]. Then, the linear transformation matrix is given by:

$$T_{ZF} = \widehat{\mathbb{H}}_n{}^{+} \tag{30}$$

where $\widehat{\mathbb{H}}_n{}^{+} = \left(\widehat{\mathbb{H}}_n^H \widehat{\mathbb{H}}_n\right)^{-1} \widehat{\mathbb{H}}_n^H$ is the pseudo-inverse of dimenion $(N_t \times N_r)$.

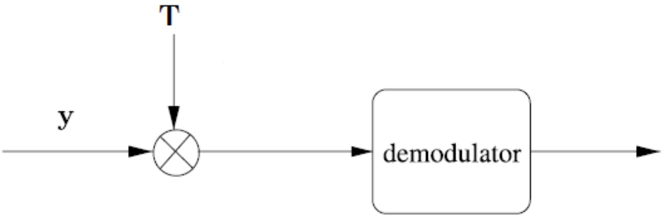


Figure 2. Conceptual illustration of linear MIMO detectors [22].

5.2 Minimum Mean Square Error Detector

The Minimum Mean Square Error (MMSE) detector is a linear detector, the transformation matrix of which minimizes the mean square error between the transmit vector X and the received vector Y. The transformation matrix T_{MMSE} is given by the solution to the following minimization problem [2], [22]-[23] and [27]-[29]:

$$T_{MMSE} = \arg_{T_{MMSE}} \min E\left(\left|\left|X - T_{MMSE}Y\right|\right|_2^2\right) \tag{31}$$

where $Y = \left[Y_n^{(1)}, \cdots, Y_n^{(N_t)}\right]^T$ and $X = \left[X_n^{(1)}, \cdots, X_n^{(N_t)}\right]^T$. Finally, the transformation T_{MMSE} can be defined as:

$$T_{MMSE} = \left(\widehat{\mathbb{H}}_n^H \widehat{\mathbb{H}}_n + 2\sigma_n^2 I\right)^{-1} \widehat{\mathbb{H}}_n^H \tag{32}$$

where σ_n^2 is the noise power.

6. OSIC Signal Detection

Detectors based on interference cancelation belong to the class of non-linear detectors, where interference due to detected stream is removed in multiple stages [22], [27] and [30]. Popular interference cancelation techniques include Ordered Successive Interference Cancelation (OSIC), which is used to improve linear detector performance without increasing the complexity significantly. OSIC is known for its simplicity. Figure 3 gives an example of three spatial streams [27], [30]. Hence, based on Equation (30), The steps involved in OSIC based detection can be summarized as follows:

1. Initially, the first stream is detected using the first row vector of Equation (30),
2. After detection and slicing to produce X_1, the remaining signal in the first stage is formed by subtracting it from the received signal as $y_1 = y - \widehat{\mathcal{H}}_{\mathrm{n}}^1 X_1$.

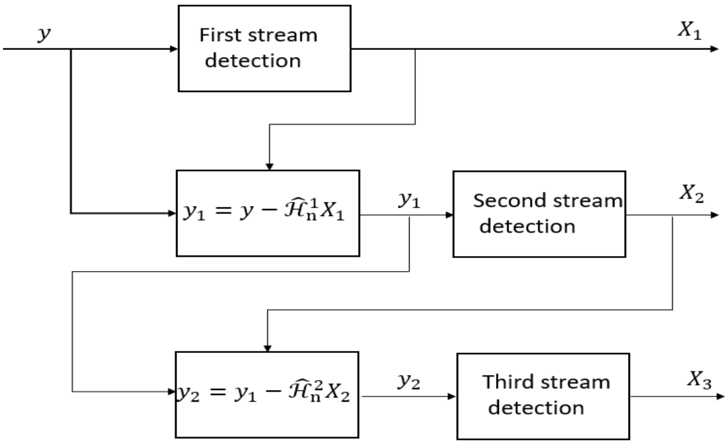

y
First stream detection
X_1
$y_1 = y - \widehat{\mathcal{H}}_{\mathrm{n}}^1 X_1$
y_1
Second stream detection
X_2
$y_2 = y_1 - \widehat{\mathcal{H}}_{\mathrm{n}}^2 X_2$
y_2
Third stream detection
X_3

Figure 3. Illustration of OSIC signal detection example of three spatial streams (i.e., N_t=3) [30].

3. The interference due to the detected stream in the first stage is canceled.
4. Another steam is detected and sliced in the second stage to produce X_2.
5. Similarly, the remaining signal and the interference in the second stage are formed by subtracting it from the received signal as $y_2 = y_1 - \widehat{\mathcal{H}}_{\mathrm{n}}^{2} X_2$.

Hence, the same processes of detection and slicing as well as interference cancelation are reproduced in the next stages [2], [24]-[27] and [30] due to the error propagation caused by erroneous decisions in the previous stages. The order of detection has significant influence on the overall performance of OSIC detection. In the next part of this section, two methods to reduce error propagation are described:

Firstly: SINR-based Ordering (SINR-BO). In this case, streams with a high post detection Signal-to-Interference-plus-Noise Ratio (SINR) are detected first [1], [27] and [30]. Based on the transformation matrix T_{MMSE}, the post-detector with SINR is defined as:

$$SINR_i = \frac{E_x |T_{i,MMSE} \widehat{\mathcal{H}}_{\mathrm{n}}^{i}|^2}{E_x \sum_{l \neq i} |T_{i,MMSE} \widehat{\mathcal{H}}_{\mathrm{n}}^{l}|^2 + \sigma_n^2 ||T_{i,MMSE}||^2} \tag{33}$$

where $T_{i,MMSE}$ is the i^{th} row of matrix (Equation 32), $\widehat{\mathcal{H}}_{\mathrm{n}}^{i}$ is the i^{th} column vector of the estimated channel matrix $\widehat{\mathbb{H}}_n$ at n^{th} time OFDM sysmbol with $i = 1, 2, \cdots, N_t$. Also, E_x is the transmitted signal energy. Furthermore, once the N_t of SINR are calculated based on Equation (32), we choose the corresponding layer with the highest SINR. In addition to that, the procedure discussed above is applied for symbol detection. Furthermore, Equation (32) is modified by the suppression of the channel gain vector equivalent to the data detected. Otherwise, the total number of SINR values to be calculated is $\sum_{i=1}^{N_t} i = \frac{N_t(N_t+1)}{2}$ [27], [30].

Secondly: SNR-based Ordering (SNR-BO). In this method, streams with a higher Signal-to-Noise Ratio (SNR) are detected first [1], [27] and [30]. Similarly based on the transformation matrix T_{ZF}, the SNR is defined as:

$$SNR_i = \frac{E_x}{\sigma_n^2 ||T_{i,ZF}||^2} \tag{34}$$

where $i = 1, 2, \cdots, N_t$. Similarly, the procedure discussed in the first method can be used. In this method, the number of SNR values to be calculated is also given by $\sum_{i=1}^{N_t} i = \frac{N_t(N_t+1)}{2}$ [1], [27] and [30].

7. SIMULATION RESULTS

In this section, a collection of performance results concerning two linear detectors (MMSE and ZF) as well as the nonlinear OSIC detectors is presented, in which their performances are evaluated in terms of BER. Hence, the computer simulation parameters are given, as shown in Table 2.

Table 2. Computer simulation parameters.

Parameters	Values
OFDM-Subcarriers	1024
OFDM symbol number (g)	10
M-QAM Modulation	64-QAM
Channel model	Rayleigh and Rician
L-taps	1, 6 and 10
$N_t \times N_r$	50×100, 50×200 and 50×300

In this part, various channel taps (i.e., 1, 6 and 10) are simulated as i.i.d. Firstly, one OFDM subcarrier (i.e., OFDM symbol) with K = 1024 (Figure 4) and a CP of ν = 256 is considered. The number of pilot tones dedicated for training is P=K/2, which are equipowered and equispaced or equipowered, equispaced and phase shift orthogonal. The length of data sequences is equal to K/2 (Figure 4). Hence, training is performed over g consecutive OFDM symbols using Monte Carlo simulation. Throughout this simulation, the number of terminals (N_t) is set to be 50 (i.e., respecting the condition treated in subsection 4.1; that is P≥LN_t).

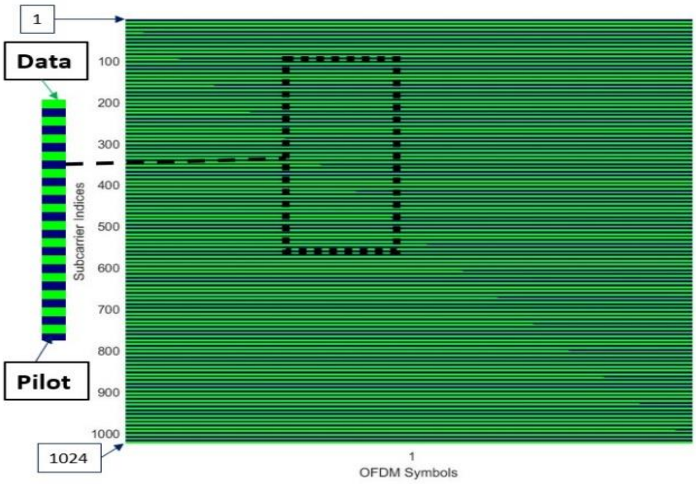


Figure 4. OFDM subcarrier for a Ma-MIMO system with $N_t = 50$ and $N_r = 100$ transmit and receive antennas, respectively.

In other words, Figure 5 shows the distributions for Rayleigh and Rician fading channels. It also demonstrates an example for K = -40 dB. The Rician distribution approaches the Rayleigh distribution and for K =15 dB, the Rician distribution approaches the Gaussian distribution. In the remainder of this paper, we assume K= -40 dB for the Rayleigh fading channel and K ≥ 15dB for the Gaussian channel.

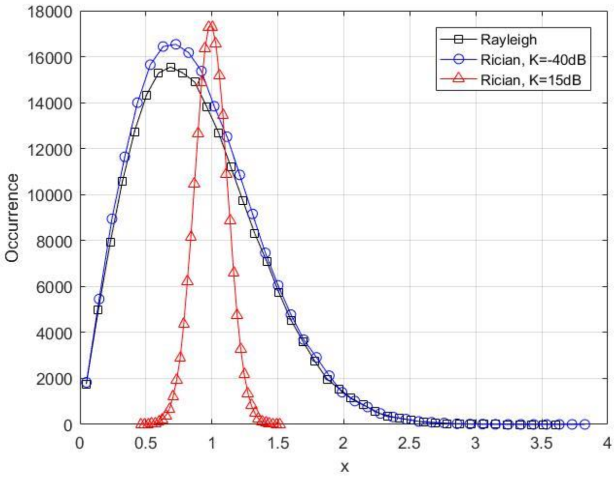


Figure 5. Distributions for Rayleigh and Rician fading channels.

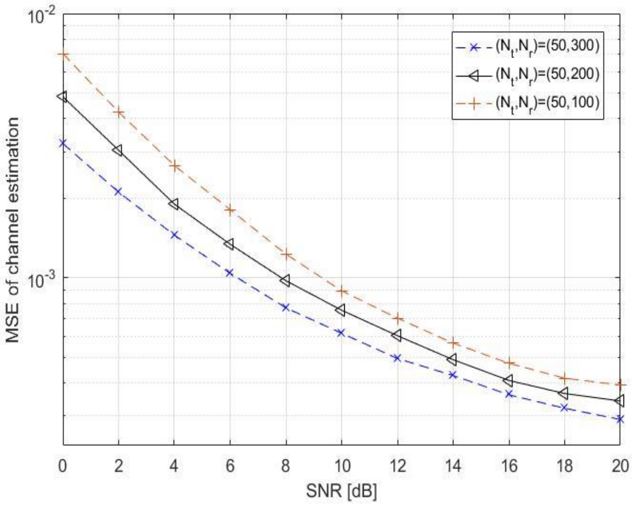


Figure 6. Channel estimation error *vs.* SNR for Rayleigh channel.

Figure 6 shows the channel estimation error for N_r = 100, 200 and 300 when Rayleigh channel is applied. The channel estimation performance improves when N_r increases and SNR increases. In related work [32], for $N_r \times N_t$= 128 × 64 Ma-MIMO system, Generalized Approximate Message Passing Detector (GAMPD) achieves an MSE of around 0.3 at SNR = 0 dB, while at SNR=2 dB, its MSE is around 0.1 for $N_r \times N_t$= 128 × 128. Moreover, for $N_r \times N_t$= 100 × 50 (Figure 6), our proposed method achieves an MSE of around 0.00705 at SNR =0 dB and 0.00424 at SNR= 2 dB. In addition, when more receive antennas are used, more spatial diversity can result in a better chance to successfully detect the data.

In the simulations, Ma-MIMO with multiple receiver antennas (N_r=100, 200 and 300) and 64-QAM modulation is used. Figure 7 shows a plot of BER *versus* SNR for L = 1 (i.e., flat fading), when a Rayleigh fading channel is employed. It is clear that the BER performance of all detectors decreases with a higher SNR. However, the MMSE performance is very close to the ZF performance. In this way, at high SNRs (i.e., small σ), MMSE behaves like ZF, since the second term inside the inverse operation

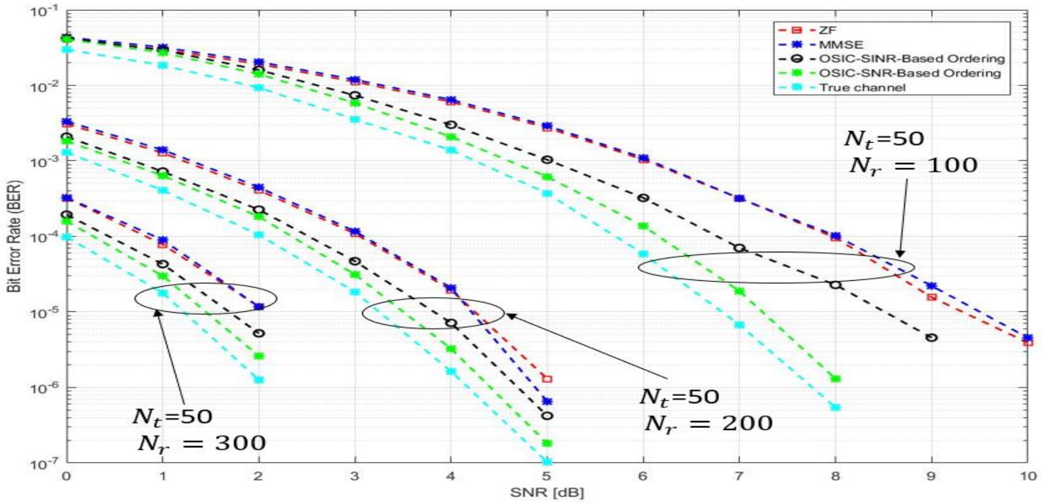


Figure 7. BER *vs.* SNR for L=1 using ZF, MMSE and various OSIC detectors for Rayleigh channel.

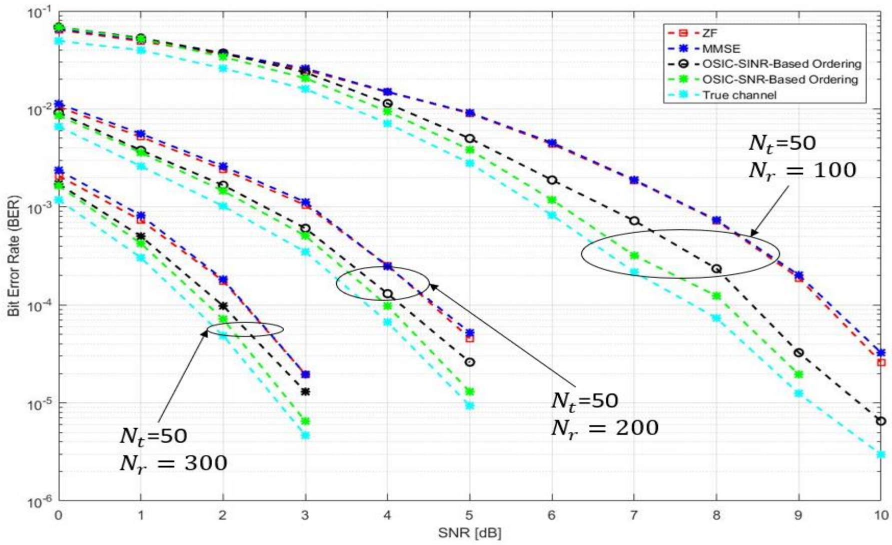


Figure 8. BER vs SNR for L=6 using ZF, MMSE and various OSIC detectors for Rayleigh channel.

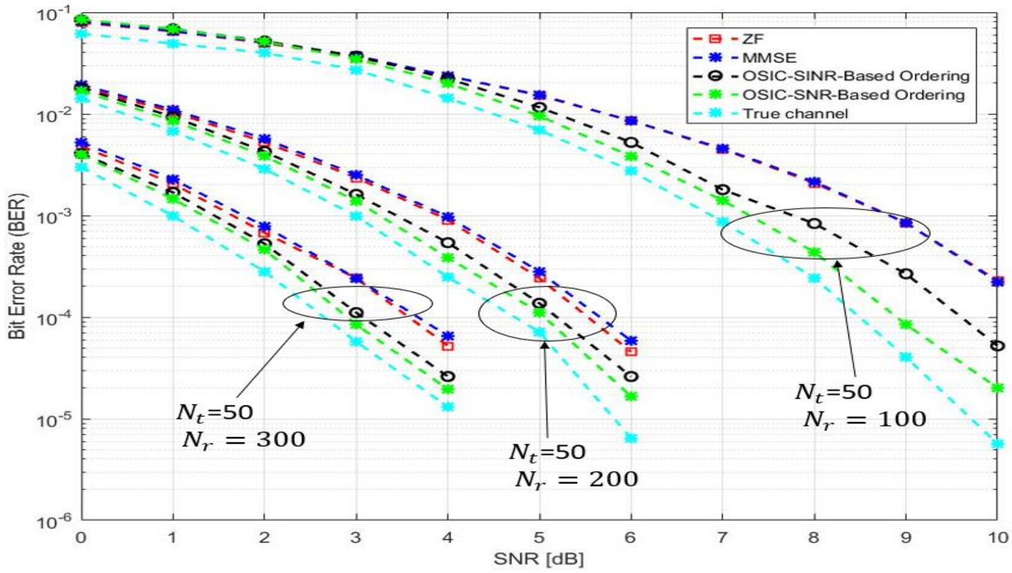


Figure 9. BER *vs.* SNR for L=10 using ZF, MMSE and various OSIC detectors for Rayleigh channel.

in Equation (32) becomes negligible. On the other hand, the performance of the OSIC detection method with SNR-BO outperforms all other detectors and is close to true channel performance. In related work [31], for $N_r \times N_t$= 128 × 16 Ma-MIMO system, refinement Jacobi (RJ)-based detector achieves a BER of around 3.2×10^{-3} at SNR = 13 dB, while for the $N_r \times N_t$ = 256 × 16 Ma-MIMO system, it achieves a BER of around 5×10^{-5} at the same SNR. For $N_r \times N_t = 100 \times 50$ Ma-MIMO, for example, ZF detector achieves a BER of around 319×10^{-6} (i.e., 0.319×10^{-3}) at SNR=7 dB. In this work and in [31], the SNR gap between the ZF and RJ-based detector at 10^{-3} BER is just about 6 dB.

Otherwise, Figures 8 and 9 show the BER performance comparison between linear and nonlinear detectors when the number of channel taps is L= 6 and 10, respectively. With an increase of channel tap number the performance of all detectors is degraded. Hence, system performance is sensitive to a higher channel taps (i.e., high frequency-selective fading). In addition to that, the gap between the OSIC-SNR-BO detector and true channel performance becomes small as the number of BS antennas increases. In related work [32], for $N_r \times N_t$= 128 × 80 Ma-MIMO system, Generalized Approximate Message Passing Detector (GAMPD) achieves a BER of around 10^{-2} at SNR = 4.3 dB. For $N_r \times N_t = 100 \times 50$ Ma-MIMO, for example, ZF detector achieves an SNR=5.74 dB at the same BER. Thus, the SNR gap between the ZF detector and GAMPD detector is just about 1.44 dB. In addition, ZF detector achieves a better performance and surpasses the GAMPD detector with a gap of 3.315 dB at BER of 10^{-2} for a Ma-MIMO of $N_r \times N_t = 200 \times 50$. Moreover, at $N_r = 300$, ZF detector is close to approximate message passing (AMP) algorithms [33], with a gap of 0.06 dB, despite a high system sensitivity to noise (i.e., 64-QAM) and high frequency-selective fading channel. The OSIC-SNR-BO detector is useful for a high multi-path fading channel or equivalently, a frequency-selective fading channel.

In addition to that, Figure 10 presents the BER in a flat Rician channel fading in the case of K = 15 (i.e., LoS) and the number of antennas at the BS equal to 100. The simulation result illustrates the convergence of all detectors over a high range of SNR and provides a bad performance. Despite the presence of multi-path effects, the system took the thermal noise from the receiver as a single source of noise (i.e., Gaussian channel). In addition, the channel behaves as the simplest statistical channel from an implementation, but not necessarily the most realistic one. However, when the BS antennas increase to 300, the system performance for the same range of SNR is improved. This improvement is likely due to the potential diversity at the BS and at the ordering (selection) in OSIC. In the case of K = -40 (NLoS), the BER decreases more and is close to Rayleigh performance. When the BS antennas increase to 300, the simulation result shows the best performance at a smaller range of SNR. Thereby, the OSIC-SNR-BO usually provides the best performance and is close to the true channel. In the same way, we consider the case of frequency-selective fading (i.e., L = 6 and 10, Figures 11 and 12, respectively). From these figures, we can see that the BER is also sensitive to the channel taps and the Rician factor. A higher number of channel taps provides a higher BER at both K= 15 and -40. Similarly, when the BS antennas

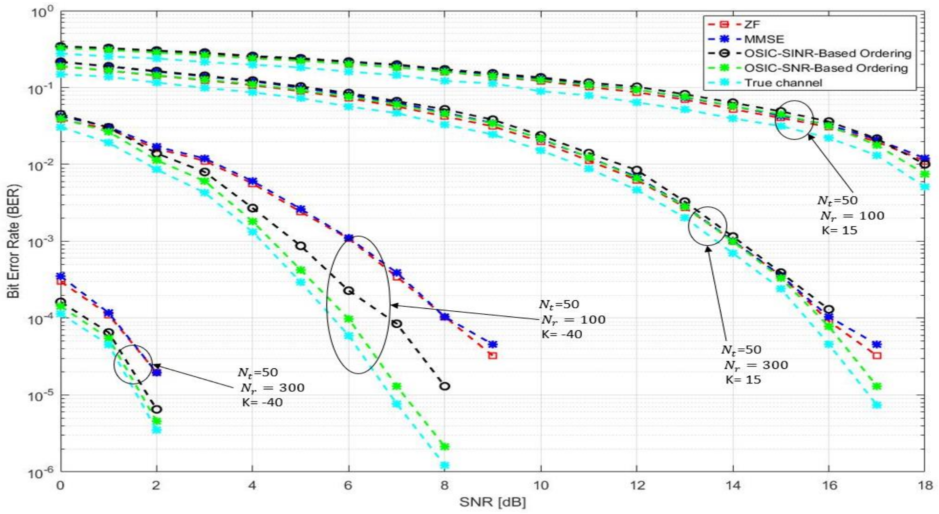


Figure 10. BER *vs.* SNR for L=1 using ZF, MMSE and various OSIC detectors for Rician channel with K= -40 and 15.

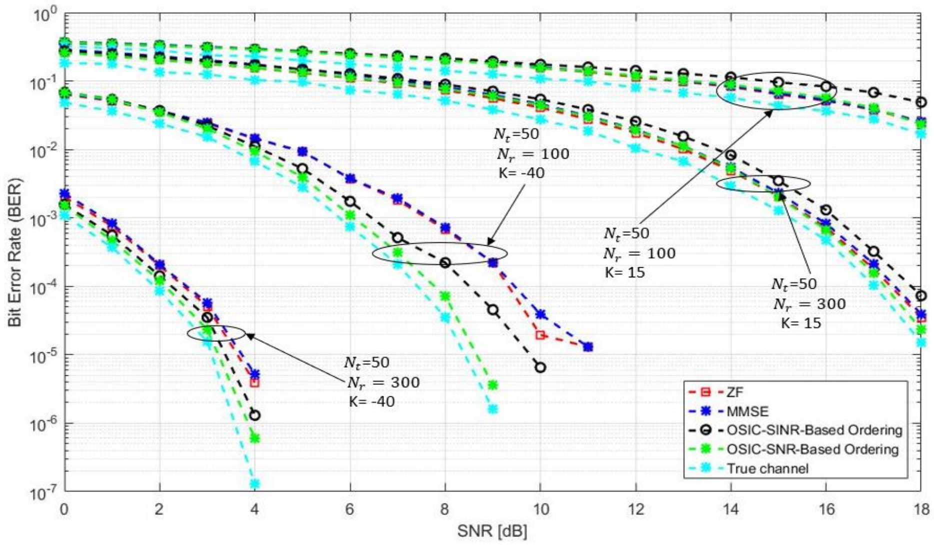


Figure 11. BER *vs.* SNR for L=6 using ZF, MMSE and various OSIC detectors for Rician channel with K= -40 and 15.

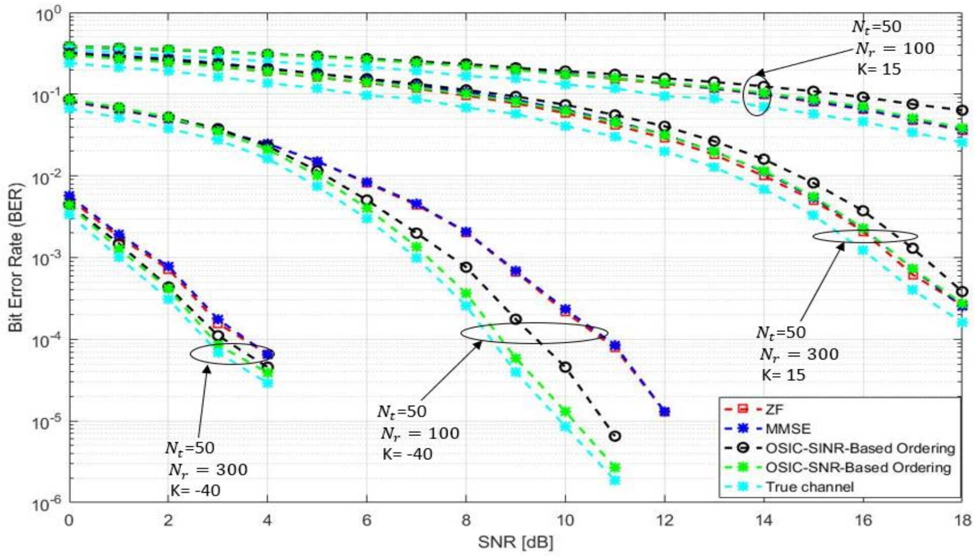


Figure 12. BER *vs.* SNR for L=10 using ZF, MMSE and various OSIC detectors for Rician channel with K= -40 and 15.

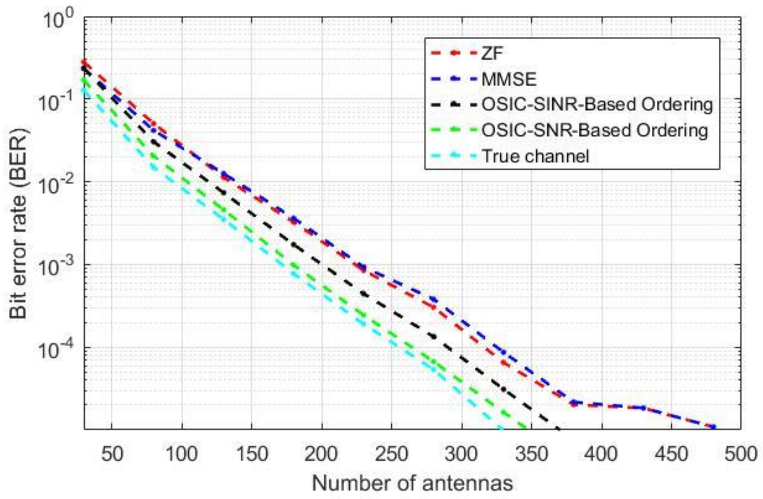


Figure 13. BER *vs.* number of antennas at the base station for L=10 using ZF, MMSE and various OSIC detectors for Rayleigh channel.

are equal to 300, the BER decreases more. Under a lower Rician factor, the system took into account the channel fading. In addition, the antenna diversity at the reception favors a constructive signal overlay. Thus, the system performance is improved and frequency-selective fading effect is compensated. In addition to that, OSIC-SNR-BO provides a better performance which is close to true channel performance.

Furthermore, Figure 13 presents the BER performance of linear detectors and nonlinear detectors with the number of antennas at the BS. The number of channel taps is equal to 10 and the SNR is set to be with a mean value 8 dB. It is clearly shown that BER decreases over a high number of antennas. In addition to that, it is observed that MMSE detector achieves a performance which is close to the performance achieved by the ZF detector. For more than 300 antennas, OSIC-SNR-BO performs slightly better than OSIC-SINR-BO due to high interference (i.e., $N_t = 50$). More interestingly, this improved performance of OSIC-SNR-BO compared to that of OSIC-SINR-BO increases remarkably as N_r increases.

8. CONCLUSIONS

In this paper, we have successively evaluated the LSCE performance in UL transmission. ZF, MMSE and OSIC detectors for a Ma-MIMO system are combined with high-order modulation 64-QAM and OFDM technique. The performance of a Ma-MIMO system in Rayleigh and Rician channel fading is analyzed with channel taps and Rician factor. The presence of many fading phenomena characterized by flat fading and frequency-selective fading degrades the system performance. Joining a large number of antennas at the BS with equipowered, equispaced and phase shift orthogonal pilot sequences can achieve a very low BER. At a higher number of antennas at the BS, the OSIC-SNR-BO provides a better performance which is close to true channel performance, while the effect of frequency-selective fading is canceled in a lower signal to noise ratio.

REFERENCES

[1] E. G. Larsson, O. Edfors, F. Tufvesson and T. L. Marzetta, "Massive MIMO for Next Generation Wireless Systems," IEEE Communications Magazine, vol. 52, no. 2, pp. 186-195, 2014.

[2] A. Chockalingam and B. Sundar Rajan, Large MIMO Systems, Cambridge University Press, 2014.

[3] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors and F. Tufvesson, "Scaling up MIMO: Opportunities and Challenges with Very Large Arrays," IEEE Signal Processing Magazine, vol. 30, No. 1, pp. 40-60, 2013.

[4] S. Singh et al., "Review of DFT-based Channel Estimation Techniques in OFDM over Multipath Channel," International Journal of Recent Research Aspects, vol. 3, no. 4, pp. 8-11, December 2016.

[5] N. Jalden, P. Zetterberg, B. Ottersten, A. Hong and R. Thoma, "Correlation Properties of Large-scale Fading Based on Indoor Measurements," Proc. of IEEE Wireless Communications and Networking Conference (WCNC 2007), pp. 1894-1899, Kowloon, China, 2007.

[6] A. Ashikhmin, T. L. Marzetta and L. Li, "Interference Reduction in Multi-cell Massive MIMO Systems I: Large-scale Fading Precoding and Decoding," IEEE Transactions on Information Theory, [Online], Available: https://arxiv.org/abs/1411.4182, November 2014.

[7] F. A. P. de Figueiredo, F. A. C. M. Cardoso, I. Moerman and G. Fraidenraich, "Channel Estimation for Massive MIMO TDD Systems Assuming Pilot Contamination and at Fading," EURASIP Journal on Wireless Communications and Networking, vol. 2018, no. 1, Article no. 14, 2018.

[8] E. Ohlmer and G. Fettweis, "Linear and Non-linear Detection for MIMO- OFDM Systems with Linear Pprecoding and Spatial Correlation," Proc. of IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, Sydney, Australia, 2010.

[9] C. Shen, Y. Zhu, S. Zhou and J. Jiang, "On the Performance of V-BLAST with Zero-forcing Successive Interference Cancellation Receiver," Proc. of IEEE Global Telecommunications Conference (GLOBECOM'04), vol. 5, pp. 2818-2822, Dallas, TX, USA, 2004.

[10] J. Han, X. Tao and O. Cui, "Lower Bound of BER in M-QAM MIMO System with Ordered ZF-SIC Receiver," Proc. of the 69th IEEE Vehicular Technology Conference (VTC), pp. 1-5, Barcelona, Spain, 2009.

[11] S. M. Maung, S. Hajime and S. Iwao, "Reduced Complexity Scheme for MIMO Receiver with Combined ZF-OSIC and ML Detection," Proc. of IEEE Symposium on Computers & Informatics (ISCI), pp. 92-96, Penang, Malaysia, 2012.

[12] J. Wang, O. Y. Wen and S. Li, "Soft-output MMSE OSIC MIMO Detector with Reduced-complexity Approximations," Proc. of the 8th IEEE Workshop on Signal Processing Advances in Wireless Communications, pp. 1-5, Helsinki, Finland, 2007.

[13] J. Wang and S. Li, "Soft-output MIMO MMSE OSIC Detector under MMSE Channel Estimation," Proc. of the 3rd IEEE International Conference on Communications and Networking in China, pp. 1117-1121, Hangzhou, China, 2008.

[14] Y. Song, C. Liu and F. Lu, "The Optimal MMSE-based OSIC Detector for MIMO System," IEICE Transactions on Communications, vol. 99, no. 1, pp. 232- 239, 2016.

[15] H. Q. Ngo, E. G. Larsson and T. L. Marzetta, "Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems," IEEE Transactions on Communications, vol. 61, no. 4, pp. 1436-1449, 2013.

[16] Y. S. Cho, J. Kim, W. Y. Yang and C. G. Kang, MIMO-OFDM Wireless Communications with MATLAB, John Wiley & Sons, 2010.

[17] P. N. Murthy and R. V. S. Satyanarayana, "A Comparison of Rayleigh and Rician Fading Channels under Frequency-selective Fading," IUP Journal of Electrical & Electronics Engineering, vol. 3, no. 4, 2010.

[18] A. Moradi, H. Bakhshi and V. Najafpoor, "Pilot Placement for Time-varying MIMO OFDM Channels with Virtual Subcarriers," Communications and Network, vol. 3, no. 1, pp. 31-38, 2011.

[19] I. Barhumi, G. Leus and M. Moonen, "Optimal Training Design for MIMO OFDM Systems in Mobile Wireless Channels", IEEE Transactions on Signal Processing, vol. 51, no. 6, pp. 1615-1624, 2003.

[20] C. D. Cantrell, Modern Mathematical Methods for Physicists and Engineers, Cambridge University Press, 2000.

[21] T.-L. Tung, K. Yao and R. E. Hudson, "Channel Estimation and Adaptive Power Allocation for Performance and Capacity Improvement of Multiple-antenna OFDM Systems," Proc. of the 3rd IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC'01), pp. 82-85, Taiwan, China, 2001.

[22] S. Yang and L. Hanzo, "Fifty Years of MIMO Detection: The Road to Large-scale MIMOs," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 1941-1988, 2015.

[23] P. Rajeev, A. Prabhat and L. Norsang, "Sphere Detection Technique: An Optimum Detection Scheme for MIMO System," International Journal of Computer Applications, vol. 100, no. 2, pp. 25-29, 2014.

[24] G. J. Foschini, "Layered Space-time Architecture for Wireless Communication in a Fading Environment when Using Multi-element Antennas," Bell Labs Technical Journal, vol. 1, no. 2, pp. 41-59, 1996.

[25] G. D. Golden, C. J. Foschini, R. A. Valenzuela and P. W. Wolniansky, "Detection Algorithm and Initial Laboratory Results Using V-BLAST Space-time Communication Architecture," Electronics Letters, vol. 35, no. 1, pp. 14-16, 1999.

[26] P. W. Wolniansky, G. J. Foschini, G. D. Golden and R. A. Valenzuela, "V-BLAST: An Architecture for Realizing Very High Data Rates over the Rich-scattering Wireless Channel," Proc. of the IEEE Conference URSI International Symposium on Signals, Systems and Electronics (ISSSE 98), Cat. No.98EX167, pp. 295-300, Pisa, Italy, 1998.

[27] A. Riadi, M. Boulouird and M. M. Hassani, "ZF/MMSE and OSIC Detectors for Upink OFDM Massive MIMO systems," Proc. of the IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 767–772, 9-11 April 2019, Amman, Jordan, 2019.

[28] A. Riadi, M. Boulouird and M M.Hassani, "Least Squares Channel Estimation of an OFDM Massive MIMO System for 5G Wireless Communications," Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT’18), vol. 2, pp. 440-450, Springer, Cham, 2018.

[29] A. Riadi, M. Boulouird and M. M’Rabet Hassani, "Performance of Massive-MIMO OFDM System with M-QAM Modulation Based on LS Channel Estimation," Proc. of the 3rd International Conference on Advanced Systems and Emergent Technologies (IC ASET’2019), Hammamet, Tunisia, 19-22 March 2019.

[30] A. Riadi, M. Boulouird and M M. Hassani, "3-D Polarized Channel Modeling for Multipolarized UCA-

Massive MIMO Systems in Uplink Transmission," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 03, pp. 231-243, December 2019.

[31] J. Minango and A. C. Flores, "Low-complexity MMSE Detector Based on Refinement Jacobi Method for Massive MIMO Uplink," Physical Communication, vol. 26, pp. 128-133, 2018.

[32] S. Wang, Y. Li, M. Zhao and J. Wang, "Energy Efficient and Low-complexity Uplink Transceiver for Massive Spatial Modulation MIMO," IEEE Transactions on Vehicular Technology, vol. 64, no. 10 , pp. 4617-4632, 2014.

[33] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang and Q. Guo, "Low-complexity Iterative Detection for Large-scale Multiuser MIMO-OFDM Systems Using Approximate Message Passing," IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 5, pp 902-915, October 2014.

[34] M. Aldababseh and A. Jamoos, "Estimation of FBMC/OQAM Fading Channels Using Dual Kalman Filters," The Scientific World Journal, vol. 2014, Article ID 586403, 2014.

[35] O. E. Ijiga, O. O. Ogundile, A. D. Familua and D. J. J. Versfeld, "Review of Channel Estimation for Candidate Waveforms of Next-generation Networks," MDPI Electronics, vol. 8, no. 9, Article no. 956, 2019.

[36] K. E. Baddour and N. C. Beaulieu, "Autoregressive Modeling for Fading Channel Simulation," IEEE Transactions on Wireless Communications, vol. 4, no. 4, pp. 1650-1662, July 2005.

ملخص البحث:

في هذه الورقة، يتم البحث في تخمين القنوات بطريقة المربعات الصغرى (LSCE) في الأنظمة الضخمة متعددة المداخل متعددة المخارج القائمة على النغمات الإرشادية. وقد تم أخذ الإرسال المعتمد على الربط العلوي (UL) بعين الاعتبار، بحيث جرى اقتراح طريقةٍ لتخمين القنوات عبر تكوين معادلة في هيئة مصفوفة تشتمل على جميع متغيرات القنوات غير المعروفة في متّجهٍ واحد وتخمينه باستخدام طريقة المربعات الصغرى. وقد تم حساب متوسط الخطأ التربيعي (MSE) لتخمين القنوات بطريقة المربعات الصغرى (LSCE).

وقد جرى تقييم الخُفوت المستوي والخُفوت المعتمد على الإنتقائية الترددية لرمزٍ واحد ولرموز متعددة من رموز الإرسال المضاعف المتعامد القائم على التقسيم الترددي (OFDM) فيما يتعلق بمتوسط الخطأ التربيعي (MSE). كما تم استقصاء المتطلب الخاص بتتابع العلامات الإرشادية في كلٍّ من الخُفوت المستوي والخُفوت المعتمد على الإنتقائية الترددية. الى جانب ذلك، اتّضح أنّ عدد العلامات الإرشادية يؤدي الى وضع تسوياتٍ مرغوبة بين هوائيّ المحطة الأساسية وتفرُّعات القنوات. وقد جرى اعتبار خُفوت القنوات المعروف بخُفوت قنوات رايلي (Rayleigh) وخُفوت قنوات ريسيان (Rician) لتقييم أداء النظام باختلاف عدد تفرُّعات كل قناةٍ من القنوات. وتم تقييم الأداء بناءً على قيم معدل الخطأ في البِتّات (BER). علاوة على ذلك، من أجل تحسين أداء الكواشف الخطية، تم استخدام كواشف غير خطية. وتبين أن متطلب تتابع العلامات الإرشادية وزيادة تنوع الاستقبال تؤدي الى تقليل قيمة معدل الخطأ (BER).

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

MULTI-LABEL RANKING METHOD BASED ON POSITIVE CLASS CORRELATIONS

Raed Alazaidah[1], Farzana Kabir Ahmad[2], Mohamad Farhan Mohamad Mohsin[2] and Wael Ahmad AlZoubi[3]

(Received: 19-Jun.-2020, Revised: 8-Aug.-2020, Accepted: 30-Aug.-2020)

ABSTRACT

Multi-label classification is a general type of classification that has attracted many researchers in the last two decades due to its applicability to many modern domains, such as scene classification, bioinformatics and text classification, among others. This type of classification allows instances to be associated with more than one class label at the same time. Class label ranking is a crucial problem in multi-label classification research, because it directly impacts the performance of the final classifiers, as labels with high ranks get a higher chance of being applied. This paper presents a new multi-label ranking algorithm called Multi-label Ranking based on Positive Correlations among labels (MLR-PC). MLR-PC captures positive correlations among labels to reduce the large search space and assigns the true rank per class label for multi-label classification problems. More importantly, MLR-PC utilizes novel problem transformation methods that facilitate exploiting accurate positive correlations among labels. This improves the predictive performance of the classification models derived. Empirical results using different multi-label datasets and five evaluation metrics reveal that the MLR-PC is superior to other commonly existing classification algorithms.

KEYWORDS

Prediction, Machine learning, Multi-label ranking, Multi-label classification, Problem transformation methods, Class ranking methods.

1. INTRODUCTION

Classification is a vital task in supervised learning that has attracted many researchers in the last few decades [1]. Classification learns rules for allocating instances to a class from a "training set" that has explicit classes. It then classifies new instances according to those rules. The accuracy of classification can be assessed by doing this to a "test set" for which the classes are known, but are not used in the classification [2].

In general, according to [3], classification problems are divided into two main categories: single-label classification (SLC) and multi-label classification (MLC). The former necessitates one class label per training instance, while the latter allows multiple class labels per instance. Thus, class labels in the SLC problems are always mutually exclusive [4], whereas class labels in MLC are not. Labels in MLC possibly have some kind of correlation [5].

In MLC problems, the task of Label Ranking (LR) is essential. It reveals the significance and the worthiness of each class label in the prediction phase. Hence, allocating each class label to its true rank is crucial. A common way to accomplish LR is to rank the available class labels according to their frequencies or probabilities [6].

One main challenge of MLC problems is the large problem search space, especially when there are large numbers of class labels and high-dimensional datasets [3]. For example, when the MLC problem contains 20 class labels, then the problem search space consists of 220 possibilities, which is computationally not cost-effective. Hence, cutting down the search space becomes a requirement.

A number of promising research attempts have been conducted in the last few years to reduce the large search space of MLC problems (i.e., [7]-[9]). These approaches dealt with MLC problems through capturing and exploiting the correlations among labels. However, many of these research studies suffer from drawbacks, mainly the limited extent of the type of correlations among class labels being captured

1. R. Alazaidah, is with Computer Science Department, Faculty of Science and IT, Jadara University. Email: raeddiab@yahoo.com

2. F. Ahmad, and M. Mohsin are with School of Computing, Universiti Utara Malaysia, Malaysia. Emails: farzana58@uum.edu.my, Farhan@uum.edu.my

3. W. Alzoubi is with Applied Science Department, Ajloun Uni. College, Al-Balqa Applied Uni., Jordan. Email: wa2010@bau.edu.jo

and adopting inefficient search techniques when capturing the correlations [10].

In this paper, a new MLR algorithm is proposed that utilizes novel problem transformation methods and reveals the positive pairwise correlations among existing labels. Considering the positive pairwise correlations among labels as a transformation criterion will facilitate the capturing and exploiting of the most accurate high-order correlations among labels. In addition, to build classification models, the proposed algorithm integrates class association rules derived by the predictive association rule mining algorithm to determine significant positive correlations among class labels. This process ensures that any negative correlations among classes are discarded (more details are given in sub-section 3.1).

The rest of the paper is organized as follows: Section 2 reviews the literature relevant to MLC, while Section 3 presents the proposed algorithm. Section 4 discusses the evaluation of the proposed algorithm and finally, Section 5 concludes and recommends future work.

2. Literature Review

MLC is a challenging problem that has attracted several scholars in the last two decades. At first, it was motivated by two domains: text classification [11] and medical diagnosis [12]-[13]. After that, MLC has been applied in several other domains, such as: automatic image and video annotation [14]-[16], classification of songs according to the emotions they invoke [17], gene functionality detection [18]-[20], protein functionality detection [21]-[22], drug discovery [23], social network mining [24]-[25], direct marketing [26] and Web mining [27].

Two main approaches have been utilized in dealing with the problem of MLC. The first approach attempts to fit a multi-label dataset into a single label classifier by transforming the multi-label dataset into one or more single-label datasets [28]. This approach has been known as PTM. The second approach adapts a single-label classifier to handle a multi-label dataset and is called the Algorithm Adaptation Method (AAM). According to [29], PTMs are preferable over AAMs, because they are easier to understand and are not domain-specific. This paper deals with the MLC problem from a PTM perspective.

Several PTMs can be found in the literature such as simple selection transformation methods. Simple selection transformation methods transform a multi-label dataset into a single-label dataset based on using the frequency of labels as a transformation criterion [30]. Hence, a multi-label instance can be transformed to be linked with the Most Frequent Label (MFL) or the Least Frequent Label (LFL). Other simple selection transformation methods ignore any multi-label instance or simply choose one of the labels that are associated with an instance randomly. This has been stated by several researchers as one of the best ways to reduce the large problem search space of the MLC problem [7], [9], [31], [52]. Therefore, the proposed algorithm in this research adopts simple PTMs that are based on positive pairwise correlations among labels, which is expected to maximize the utilization of the most significant positive correlations among labels (See Section3 for further details).

In [51], the authors questioned the usefulness of using simple transformation methods that are based on label frequency. Therefore and in order to maximize the exploitation of the most accurate positive dependencies among labels, they proposed three novel simple problem transformation methods based on the positive dependencies among labels and not based on the frequency of labels as in the traditional transformation methods. The first transformation method has been dubbed HAPCF, short for High Accurate Positive Correlation First, where the pairwise positive correlations for the labels are captured and then, the labels are ordered in a descendent way according to the accuracy of the high accurate positive correlation for each label. The second transformation method is called High Standard Deviation First (HSDF), where the Standard Deviation for the accuracy of the pairwise positive correlations for each label is calculated and then, the labels are ordered in a descendent way. The third transformation method is a hybrid method of the first and the second method and has been called High Accurate Positive Correlation and Standard Deviation First (HAPCSDF).

The proposed transformation methods have been extensively evaluated using seven different multi-label datasets and five evaluation metrics, where they showed a superior performance compared with the existing transformation methods. The authors concluded that utilizing the correlations among labels as a transformation criterion is better than using the frequency of labels as a transformation criterion. Furthermore, according to the degree of the captured correlations among labels, MLC algorithms could be categorized into three types. The first type is known as the first-order approach, which ignores any correlations among labels. Hence, labels in the first-order approach are considered as mutually exclusive. This approach has the advantage of being simple, but suffers from low predictive performance, especially with large datasets that have a high number of labels [32]. Examples of the first-order MLC algorithms are: the Binary Relevance and the Multi-label K Nearest Neighbor (ML-KNN) algorithm [33].

The second type is called the second-order approach and depends on extensive pairwise comparisons among labels while considering features values [3]. This approach suffers from a limited ability to capture correlations among labels as well as the substantial number of the pairwise comparisons that are needed. Thus, the second approach is unsuitable for datasets with a high number of labels [3]. Examples of the second-order approach algorithms are: the Ranking by Pairwise Comparisons (RPC) algorithm [34] and the Calibrated Label Ranking (CLR) algorithm [35].

The third type of MLC algorithm according to the degree of correlations being captured is the high-order approach. This approach captures high-order correlations among labels in the whole dataset or among a subset of the dataset [32]. Usually, this approach suffers from a high-complexity issue due to utilizing complex techniques to capture the correlations among labels. Nevertheless, the high-order approach tends to be better than the previously discussed two approaches, especially for datasets with high cardinality [3]. Examples of high-order approach algorithms are: Label Powerset (LP) [36], Pruned Set (PS) and Ensemble of Pruned Set (EPS) [37], RAKEL [38], Classifier Chains (CC) and Ensemble of Classifier Chains (ECC) [29] and MLC-ACL [30].

Several algorithms that belong to different learning strategies have been proposed to solve the problem of MLC. In [36], an algorithm called HOMER was presented. HOMER is short for Hierarchy of Multi-label classifiERs. HOMER aims to handle large datasets by using a tree structure. It is a divide-and-conquer-based algorithm that constructs a tree recursively in a top-down, depth-first fashion, starting from the root. The HOMER algorithm has been evaluated using two large datasets (delicious and mediamill) and compared with the BR method. HOMER outperforms BR in prediction accuracy, running time and scalability to large datasets. However, HOMER needs to be evaluated against different algorithms and methods and not only the BR method. The complexity of HOMER could be reasonable when applied to large datasets, but HOMER will be inefficient when applied to small or moderate datasets. HOMER is more suitable to large datasets with a large number of labels.

Zhang and Wu (2015) [39] questioned the usefulness of using the same traditional feature selection methods with MLC and, based on their reflections, they proposed an algorithm that focused on extracting label specific features. The algorithm has been named as Multi-label Learning with Label-specific Features (LIFT). LIFT starts by applying clustering techniques on each label to determine its positive and negative instances. Then, features that are specific to each label in the label set are constructed by using the positive instances that were found in the first step. Finally, (k) classifiers are used in the training step. Each classifier is trained using the specific features that were generated previously and for every label in the label set.

LIFT was evaluated using eight datasets from different domains and compared to several state-of-the-art algorithms. The evaluation process concluded that the effectiveness of using new feature-selection techniques was more suitable to the nature of the MLC problem. LIFT has the advantage of being a general approach that could be used with any multi-label algorithm as a preprocessing step that may enhance the effectiveness and the efficiency of the algorithm. On the other hand, LIFT ignores any correlations among labels in the process of selecting the features.

Back Propagation for Multilabel Learning (BP-MLL) algorithm [26] is an adaptation of the traditional multi-layer, feed-forward neural network to multi-label data. The net was trained with a gradient descendent and error back propagation with an error function closely related to the ranking loss that took into account the multi-label data. Experimental results showed a competitive performance in genomics and text categorization domains, with a computational cost derived according to neural network methods. Rokach, Schclar and Itach (2014) [40] questioned the usefulness of selecting the subsets randomly in RAKEL. Their view was based on the idea that dividing the original label sets into smaller subsets should be considered wisely and not randomly. These subsets should reserve the inter-label correlations and other constraints. The chosen (k) subsets should cover all labels and be the minimum possible. The authors proposed using approximation algorithms to determine the size and contents of the subsets. They proved the efficiency of their algorithms by using different evaluation metrics and different datasets. The only limitations of their work were high complexity and running time [41].

LR methods handle the problem of Multi-label Learning (MLL) by transforming it into a problem of ranking, where pairwise comparisons are performed among all labels and based on these comparisons, a final ranking is obtained. Two main popular methods that could be found in the literature of MLL that are based on pairwise comparisons. The first method is called Ranking by Pairwise Comparisons (RPC) [34]. RPC is similar to BR in dividing a dataset with (k) labels into (k (k-1)/2) binary datasets; a binary dataset for each pair of labels (L1, L2), where the instances of the dataset are those instances that are associated with L1 or L2, but not both labels. To classify a new instance, all the binary models are invoked and a ranking is obtained by counting the votes for each label. RPC suffers from several limitations, such as high quadratic complexity that makes it a very bad choice when dealing with a large number of labels. The last limitation of RPC is that it does not have a split point between relevant and irrelevant labels [3].

CLR method is another pairwise method that enhanced RPC by introducing a calibration label. This virtual label (L0) works as a split point between relevant labels and irrelevant labels [35]. As in RPC, the CLR method suffers from space complexity and computational complexity as well [3].

3. The Proposed Algorithm: Multi-Label Ranking Algorithm based on Positive High-order Correlations among Labels (MLR-PC)

The MLR-PC algorithm comprises three main phases: the transformation phase, the multi-label classifier construction phase and the prediction phase. In the first phase, MLR-PC transforms a multi-label dataset into a single-label dataset using a transformation method based on the positive pairwise correlations among labels and applies a rule-based classifier on the transformed dataset to construct a single-label classifier. The PART algorithm [42] has been chosen in this paper as a base classifier. In the second phase, a multi-label classifier is constructed based on the discovered high-order positive correlations among labels that respect the transformation order of the labels. The last phase involves assigning the predicted ranked labels to a test instance. The main steps of the MLR-PC algorithm are shown in Algorithm 1 and more details are given in the following subsections.

```
Input: D - Multi-label dataset, minacc – minimum accuracy threshold
Output: mlC - multi-label Classifier
Building Model (D, minacc, TD)
{
Algorithm2 (D)
For each x ∈X in TD            // X = The set of all labels
{
  Generate all Positive Association Rules (PARs) in a form <<x ⟶ y>>, where y has a lower transformation order
        than x, using Predictive Apriori algorithm.          / /with respect to minacc
  Repeat the previous step having (x and y) in the Antecedent and z in the Consequent, where z ∈λ-{x,y}
  PARs ⟵ Algorithm3 (PAR)
  PARs ⟵ Merge (PAR), where the Antecedent is x and the consequent belongs to (λ-x)
  For each rule in S and have x in its consequent, Replace (x, PARs)
}
Return (mlC)
```

Algorithm 1. MLR-PC algorithm.

3.1 Transformation Phase

The transformation phase fits the multi-label dataset into the single-label classifier and often this step depends on the frequency of labels as a transformation criterion. For the proposed algorithm, the transformation phase relies on a new criterion that is based on the positive pairwise correlations among labels. Hence, the label space of the input multi-label dataset is first extracted and then class association rules are derived using the Predictive Apriori algorithm [43]. These rules are utilized to capture the positive pairwise correlations among labels by keeping only positive rules in the form of “IF C1=1 THEN C2=1”for further analysis regardless of the accuracy measure of the rules. In the proposed algorithm, the multi-label dataset is transformed into a single-label dataset based on one of the following transformation methods: (HAPCF, HSDF, HAPCSDF, MFL and LFL). For more clarification and information regarding the previous three PTMs, the reader is advised to read reference [51]. The following step (Step 9) aims to formulate the transformed Single-label Dataset (SLD) using one of the previous PTMs. After that, PART algorithm is trained on the formulated SLD to construct the single-label classifier (S).

The transformation phase has been performed based on the positive pairwise correlations among labels and not based on the frequency of labels. Algorithm 2 shows the transformation algorithm that has been adopted in the MLR-PC algorithm.

```
Input:    D - Multi-label dataset, minacc – minimum accuracy threshold
Output: S - Single label classifier
Transformation Phase (D)
{
1 TD ←Extract Label Space (D) // TD refers to the label space, where it is represented as a transactional dataset
2 For each item (x) in TD
3 {
4   PARs set ←   Predictive Apriori (TD)      // regardless the predictive accuracy of PARs
5   HAPCF ← Predictive Accuracy of the highest PAR
6   HSDF ← Compute the Standard Deviation among the Predictive Accuracy of the captured PARs
7   HAPCSDF ← HAPCF + HSDF
8 }
9 SLD ← Transform (D, {HAPCF, HSDF, HAPCSDF})  //SLD: the transformed Single Label Dataset
10   S ← PART (SLD)  / with respect to minacc  // S= single label classifier
11   Return (S, TD)
```

Algorithm 2. Transformation algorithm.

After transforming the input multi-label dataset into a single-label dataset, classifiers, such as PART, JRip, BayesNet and RIPPER, among others, could be used as a base classifier for the transformed dataset (SLD). In this paper, the PART algorithm has been selected as a base classifier after a thorough evaluation of different algorithms. Table 1 shows the main characteristics of the multi-label datasets considered in this paper.

Table 1. Dataset characteristics.

Dataset	Instances	Attributes	Labels	LCard	Domain
Yeast	2417	103	14	4.327	Biology
Scene	2712	294	6	1.074	Image
Emotions	593	72	6	1.868	Media
Flags	194	19	7	3.392	Image
Genbase	662	1186	27	1.252	Biology
TMC2007	28596	500	22	2.16	Text
Ohsumed	13929	1002	23	1.66	Text

To determine the best base classifier to use in MLR-PC algorithm, five different classifiers have been evaluated on the datasets shown in Table 1. These classifiers are: BayesNet [44], JRip [45], Simple Logistic [46], Decision Table [47] and PART [42]. Table 2 shows the evaluation of several rule-based single-label classifiers, on four different multi-label datasets, based on the Accuracy metric.

Two main points could be inferred from Table 2. The first point is that the transformation method being used has a significant influence on the accuracy rates of the base classifier. In general, the MFL method shows a better accuracy on three datasets (Emotions, Flags and Yeast), while LFL shows a better accuracy on the Scene dataset, which has a low cardinality that is nearly 1. The second point is that the PART algorithm achieved the best performance in terms of accuracy on all datasets regardless of the transformation method being used. Therefore, the PART algorithm has been used as a base classifier for the proposed MLR-PC algorithm.

3.2 Multi-label Classifier Construction Phase

Step 2 to Step 8 of the MLR-PC algorithm (Algorithm 1) produce multi-label rules by integrating the single-label rules (s) discovered by the PART algorithm into multi-label rules. This goal is achieved by capturing the high-order positive correlations among labels based on the positive pairwise correlations discovered earlier by Algorithm 2. For example, a positive pairwise correlation exists between labels

C1, C2 and C3, then MLR-PC attempts to discover a rule like "IF C1 and C2 THEN C3" or a rule like "IF C1 THEN C2 and C3", where C2 and C3 must have a transformation order less than the transformation order of C1. To make the building multi-label rules phase clear, suppose that the following rules have been discovered with label X in the antecedent:

Table 2. The predictive accuracy of several classifiers on the transformed versions of some multi-label datasets.

Base Classifier	**Emotions**		**Flags**		**Yeast**		**Scene**	
	MFL	**LFL**	**MFL**	**LFL**	**MFL**	**LFL**	**MFL**	**LFL**
BayesNet	72.770	63.860	75.380	41.530	71.900	30.100	77.170	82.600
JRip	82.670	72.280	81.540	46.150	80.940	45.480	78.850	78.930
Simple Logistic	73.760	73.760	81.530	52.300	71.900	73.240	80.850	86.780
Decision Table	69.800	57.430	75.380	41.540	71.910	30.770	58.860	63.380
PART	**96.530**	**96.040**	**89.230**	**84.620**	**95.650**	**92.310**	**98.410**	**98.830**

IF X THEN Y	Accuracy (0.887)
IF X THEN Y and Z	Accuracy (0.659)
IF X THEN Y and W	Accuracy (0.742)

The above three rules are ordered and merged into a single rule using Algorithm 3 as follows: IF X THEN Y, W, Z. Algorithm 3 shows the sorting procedure for the captured positive association rules. After that and for all single-label rules learned by PART algorithm, every rule that has label X as a consequent will be modified by the MLR-PC algorithm. Thus, the new consequent is Y, W, Z. The process of converting all single-label rules learned by PART continues with all other rules in the same way.

```
Input: Set of positive association rules
Output: Sorted positive association rules
For any two given rules, $r_1$ and $r_2$, $r_1$ precedes $r_2$ if:
1. The Predictive Accuracy of $r_1$ is higher than that of $r_{2.}$
2. Both rules have the same Accuracy value, but the cardinality of $r_1$ is higher than that of $r_2$.
3. Chose randomly, when the two previous conditions are the same for $r_1$ and $r_2$.
```

Algorithm 3. Ordering the positive association rules algorithm.

3.3 Prediction Phase

As a test instance is about to be classified, the prediction procedure of MLR-PC works as follows. The procedure starts with searching all over the final multi-label rules in the rule set (mlC), to find the best rule that matches the instance test (the rule's body matches some attribute values of the test instance). As the best rule that matches the instance is determined, the set of the labels of that rule is associated with the test instance in the same order as they appear in the consequent of the fired rule. This method utilizes only one rule to associate the predicted class label to a test instance.

4. Empirical Analysis of MLR-PC Algorithm

The MLR-PC algorithm has been implemented using Java and integrated into the Waikato Environment for Knowledge Analysis (WEKA) software system [48], which is an open source Java software. All experiments were carried out on a Pentium IV, Core i3, 2.10 GHz computer. The training datasets and the testing datasets were chosen according to dataset author recommendation, where nearly two thirds of the datasets have been used as training sets and one third of them been used as testing sets. All datasets are available in Mulan, a multi-label dataset repository [49].The following two sub-sections show the evaluation of the proposed algorithm. Sub-sections 4.1 and 4.2 discuss the evaluation of the MLR-PC algorithm on regular-sized and large-sized datasets, respectively.

4.1 Evaluation of the Proposed MLR-PC Algorithm on Regular-Sized Datasets

Table 3 to Table 6 show a comparison between the proposed MLR-PC algorithm and other MLL algorithms. The compared algorithms have been chosen to represent the three main MLL approaches. The first-order approach was represented by two algorithms: BR and ML-KNN. The second-order approach was represented by two algorithms: BP-MLL and CLR. Finally, the high-order approach was represented by seven algorithms: LP, RAKEL, CC, PS, ECC, EPS and ML-LOC.

Also, the chosen algorithms belong to both PTMs (BR, CLR, LP, RAKEL, CC, PS, ECC and EPS) and AAMs (ML-KNN and BP-MLL). Five multi-label evaluation metrics have been used to evaluate the proposed MLR-PC algorithm: Accuracy, Hamming Loss, Exact Match, One-error and Coverage [50], [53].

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:

$$Accuracy \quad = \frac{1}{t}\sum_{i=1}^{t} \frac{|Z_i \cap Y_i|}{|Z_i \cup Y_i|} \tag{1}$$

Hamming Loss is a multi-label classification metric that measures how many times on average an instance-label is misclassified. This metric considers both error predictions (when the wrong label is predicted) and omission errors (when the correct label is not predicted). For this metric, the lower the value, the better the accuracy and the performance of the classifier [51]. Hamming Loss is computed using the following equation, where (Δ) denotes the symmetric difference between the grounded truth label set and the predicted set.

$$Hamming\ loss = \frac{1}{t}\sum_{i=1}^{t} \frac{1}{q} |Z_i \triangle Y_i| \tag{2}$$

Exact Match is a restrict metric that does not distinguish between partially correct and completely incorrect prediction. This metric calculates the average of instances whose predicted labels are exactly the same as their grounded truth labels. Exact Match is computed using the following equation and must be maximized:

$$Exact\ Match = \frac{1}{t}\sum_{i=1}^{t} [\, Z_i = Y_i] \tag{3}$$

One-error metric calculates how many times the top–ranked label was not in the set of predicted labels. For this metric, it is clear that it is not suitable for MLL problem; since it considers only the top–ranked label and neglects all other labels. One-error metric must be minimized and is calculated using the following equation:

$$One - error = \frac{1}{t}\sum_{i=1}^{t} [\, arg\, min\, \tau_i(\lambda) \notin \ Y_i, \lambda \in \mathcal{L}] \tag{4}$$

Coverage measures the average depth in the ranking, in order to cover all the labels associated with an instance. The lower the value of the Coverage metric, the better the accuracy and the performance. This metric is more suitable than the One-error metric for MLL problem; since it considers all labels associated with the instance and not only the top–ranked label. The following equation is used to calculate the Coverage metric:

$$Coverage = \frac{1}{t}\sum_{i=1}^{t} max\, \tau_i(\lambda) - 1\,, \lambda \in Y_i \tag{5}$$

It is worth mentioning that (Y_i) represents the grounded truth label set, while (Z_i) represents the predicted label set. Also, (t) and (q) represent the total number of instances and the total number of labels in the dataset, respectively.

Four regular-size multi-label datasets have been considered in the evaluation of the proposed MLR-PC algorithm. Table 3 shows the evaluation of the proposed MLR-PC algorithm, using five PTMs (HAPCF, HSDF, HAPCSDF, MFL and LFL), based on the Accuracy metric. Table 3 shows clearly that the MLR-PC algorithm has the best accuracy among all other considered algorithms.

Table 3. Accuracy rates of the different MLL algorithms on regular-sized datasets.

	Algorithm	**Yeast**	**Scene**	**Emotions**	**Flags**
MLR-PC + PTM	MLR-PC-HAPCF	0.532	**0.908**	**0.738**	**0.671**
	MLR-PC-HSDF	**0.583**	**0.908**	**0.738**	0.617
	MLR-PC-	0.538	**0.908**	**0.738**	0.620
	MLR-PC-LFL	0.514	0.881	0.718	0.562
	MLR-PC-MFL	0.280	0.885	0.559	0.483
1st order	BR	0.520	0.643	0.551	0.576
	ML-KNN	0.520	0.691	0.366	0.555
2nd Order	BP-MLL	0.185	0.212	0.276	NG
	CLR	0.514	0.695	0.557	NG
High Order	LP	0.530	0.735	0.584	NG
	RAKEL	0.493	0.694	0.592	NG
	CC	0.521	0.736	0.584	NG
	PS	0.533	0.751	0.599	NG
	ECC	0.299	0.270	0.282	NG
	EPS	0.537	0.751	0.599	NG
	ML-LOC	0.510	NG	0.497	0.568

Table 4. Hamming Loss rates of the different considered MLL algorithms on regular-sized datasets.

	Algorithm	**Yeast**	**Scene**	**Emotions**	**Flags**
MLR-PC + PTM	MLR-PC-HAPCF	0.144	**0.001**	**0.116**	**0.174**
	MLR-PC-HSDF	**0.127**	**0.001**	**0.116**	0.187
	MLR-PC-HAPCSDF	0.143	**0.001**	**0.116**	0.187
	MLR-PC-LFL	0.158	**0.001**	0.119	0.213
	MLR-PC-MFL	0.219	**0.001**	0.149	0.266
1st order	BR	0.193	0.009	0.188	0.274
	ML-KNN	0.193	0.008	0.262	0.284
2nd order	BP-MLL	0.322	0.057	0.433	NG
	CLR	0.226	0.101	0.214	NG
High Order	LP	0.206	0.090	0.198	NG
	RAKEL	0.207	0.095	0.186	NG
	CC	0.211	0.100	0.197	NG
	PS	0.205	0.084	0.192	NG
	ECC	0.619	0.470	0.630	NG
	EPS	0.207	0.085	0.193	NG
	ML-LOC	0.193	NG	0.210	0.262

For the Scene dataset, no positive correlations among labels were discovered. Nevertheless, the accuracy of the MLR-PC algorithm on this dataset is still the highest. The reason for that is the high accuracy of the base classifier being used (PART). Also, for the Emotions dataset, same transformation orders have been discovered, when using any of the correlation-based PTMs. Hence, MLR-PC has the same accuracy using any of the correlation-based PTMs. Finally, Table 3 demonstrates that the correlation-based PTMs have greatly affected the accuracy of the proposed MLR-PC algorithm.It is worth mentioning that "NG" refers to a "Not Given" value, either because the metric is not applicable to the algorithm or it has not been provided in the original article. Also, all algorithms and methods have been considered with their default settings as stated in their original articles.

Table 4 shows the evaluation results of the proposed MLR-PC algorithm, with respect to the PTMs being used, using the Hamming Loss metric. The table shows clearly that the MLR-PC algorithm has the lowest Hamming Loss of all the algorithms. Also, the performance of the proposed MLR-PC algorithm when utilizing the correlation-based PTMs (HAPCF, HSDF and HAPCSDF) is much better than when utilizing the conventional frequency-based PTMs (MFL and LFL) on most multi-label datasets.

Figure 1 shows the Exact Match results of the proposed MLR-PC algorithm, with respect to the PTMs being used, compared with other algorithms. Figure1 shows that the proposed MLR-PC algorithm has the best Exact Match on the Emotions, Scene and Flags datasets, while PS has the best Exact Match on the Yeast dataset. In general, the proposed MLR-PC algorithm shows an excellent predictive performance, with respect to the Exact Match metric on most regular-sized datasets considered in this paper.

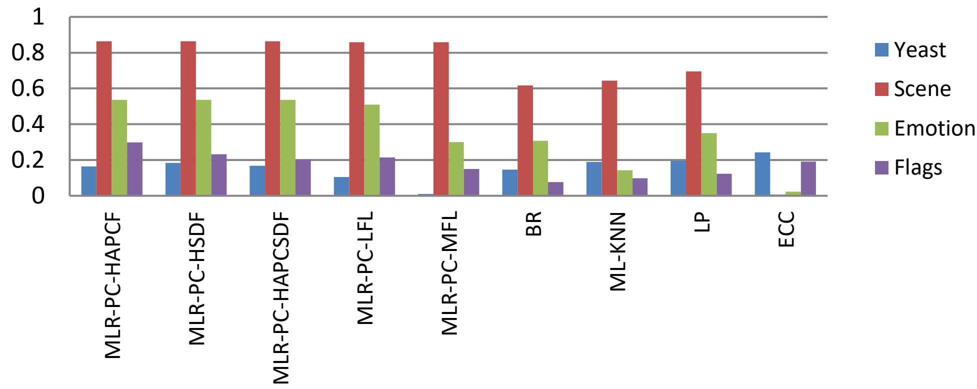


Figure 1. Evaluation of the proposed MLR-PC algorithm using the Exact Match metric on regular-sized datasets.

Table 5. One-error rates of the different considered MLL algorithms on regular-sized datasets.

	Algorithm	Yeast	Scene	Emotions	Flags
MLR-PC + PTM	MLR-PC-HAPCF	0.053	**0.055**	0.039	**0.076**
	MLR-PC-HSDF	0.076	**0.055**	0.039	0.107
	MLR-PC-HAPCSDF	0.063	**0.055**	0.039	0.107
	MLR-PC-LFL	0.076	0.061	0.039	0.153
	MLR-PC-MFL	**0.043**	0.059	**0.034**	0.107
1st order	BR	0.227	0.262	0.256	NG
	ML-KNN	0.228	0.219	0.263	NG
2nd order	BP-MLL	0.235	0.821	0.318	NG
	CLR	0.241	0.323	0.291	NG
High Order	LP	0.267	0.246	0.310	NG
	RAKEL	0.255	0.237	0.260	NG
	CC	0.256	0.268	0.283	NG
	PS	0.321	0.287	0.427	NG
	ECC	0.685	0.775	0.802	NG
	EPS	0.265	0.225	0.300	NG
	ML-LOC	NG	NG	NG	NG
	BR+	NG	NG	NG	NG

Table 5 shows the evaluation results of the proposed MLR-PC algorithm, with respect to the PTMs being used, using the One-error metric. Table 5 shows clearly that MLR-PC algorithm has the best results among all other MLL algorithms, considering the One-error metric. The main reason for that is using PART algorithm as a base classifier. PART algorithm shows a superior performance on multi-label datasets that usually have special characteristics, such as high dimensionality, high number of instances and most attributes being continuous.

Table 6 shows the evaluation of the proposed MLR-PC algorithm considering the Coverage metric with respect to the PTMs being used. It is obvious from Table 6 that MLR-PC algorithm has the best Coverage among all MLL algorithms considered in this research on the four regular-sized datasets.

Table 6. Coverage rates of the different considered MLL algorithms on regular-sized datasets.

	Algorithm	Yeast	Scene	Emotions	Flags
MLR-PC + PTM	MLR-PC-HAPCF	5.172	**0.133**	**1.193**	2.892
	MLR-PC-HSDF	4.597	**0.133**	**1.193**	2.938
	MLR-PC-HAPCSDF	5.107	**0.133**	**1.193**	**2.830**
	MLR-PC-LFL	**4.298**	0.158	1.352	3.000
	MLR-PC-MFL	5.650	0.162	1.524	3.738
1st order	BR	6.350	1.232	2.400	NG
	ML-KNN	6.300	0.456	2.320	NG
2nd order	BP-MLL	8.005	0.744	3.150	NG
	CLR	NG	NG	NG	NG
High Order	LP	8.065	0.733	2.235	NG
	RAKEL	9.155	0.593	1.986	NG
	CC	7.249	0.619	1.756	NG
	PS	8.313	0.845	2.331	NG
	ECC	10.731	2.662	3.817	NG
	EPS	8.303	0.689	2.138	NG

4.2 Evaluation of the Proposed MLR-PC Algorithm on Large-Sized Datasets

Table 7 shows the evaluation results of the proposed MLR-PC algorithm, using five PTMs (HAPCF, HSDF, HAPCSDF, MFL and LFL), based on the Accuracy metric, with respect to several other algorithms. Table 7 clearly shows that the proposed MLR-PC algorithm has a superior performance on the three large-sized datasets. Also, the proposed MLR-PC algorithm has the best accuracy values when considering the correlation-based PTMs, especially on the Genbase dataset.

Table 8 shows the evaluation results of the proposed MLR-PC algorithm on large-sized datasets, considering the Hamming Loss metric, with respect to several MLL algorithms. From Table 8, the conclusion can be made that the proposed MLR-PC algorithm has the best Hamming Loss metric on the three large-sized datasets, especially when using the correlations-based PTMs.

Table 7. Evaluation of the proposed MLR-PC algorithm using Accuracy metric on large-sized datasets.

	Algorithm	Genbase	TMC2007	Ohsumed
MLR-PC + PTM	MLR-PC-HAPCF	0.985	**0.654**	**0.741**
	MLR-PC-HSDF	**0.988**	**0.654**	**0.741**
	MLR-PC-HAPCSDF	0.987	**0.654**	**0.741**
	MLR-PC-LFL	0.981	**0.654**	**0.741**
	MLR-PC-MFL	0.929	0.635	**0.741**

1st order	BR	0.962	0.541	0.361
	ML-KNN	0.948	0.531	0.355
2nd order	BP-MLL	0.632	0.652	0.403
	CLR	0.561	0.506	0.374
High Order	RAKEL	0.982	0.549	0.383
	LIFT	NG	NG	NG
	ECC	0.978	0.517	0.426
	EPS	0.945	0.549	0.424

Table 8. Evaluation of the proposed MLR-PC algorithm using the Hamming Loss metric on large-sized datasets.

	Algorithm	Genbase	TMC2007	Ohsumed
MLR-PC + PTM	MLR-PC-HAPCF	**0.001**	**0.039**	**0.002**
	MLR-PC-HSDF	**0.001**	**0.039**	**0.002**
	MLR-PC-HAPCSDF	**0.001**	**0.039**	**0.002**
	MLR-PC-LFL	0.002	**0.039**	**0.002**
	MLR-PC-MFL	0.008	0.043	**0.002**
1st order	BR	**0.001**	0.071	0.007
	ML-KNN	0.005	0.073	0.007
2nd order	BP-MLL	0.004	0.098	0.008
	CLR	0.004	0.068	0.008
High Order	RAKEL	0.003	0.068	0.043
	LIFT	0.003	NG	0.056
	ECC	0.002	0.068	0.067
	EPS	0.007	0.069	0.074

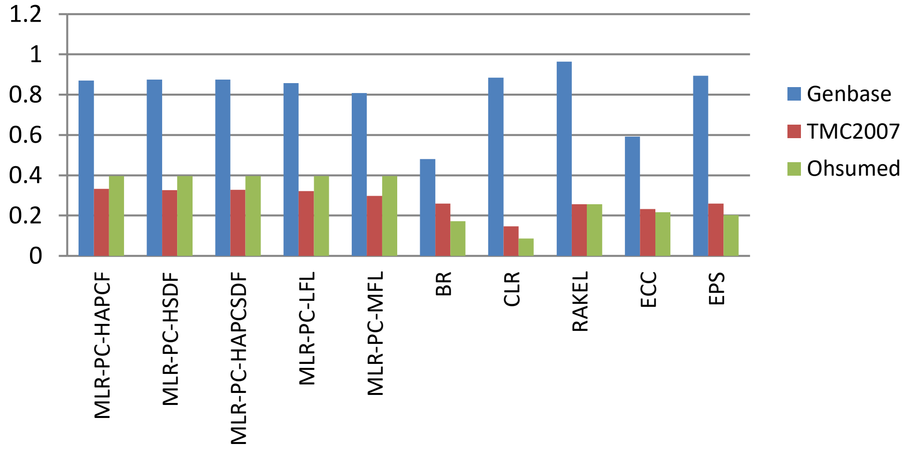


Figure 2.Evaluation of the proposed MLR-PC algorithm using the Exact Match metric on large-sized datasets.

Figure 2 shows the evaluation results of the proposed MLR-PC algorithm using the Exact Match metric, with respect to other MLL algorithms. Figure 2 clearly shows that the proposed MLR-PC has a superior performance on TMC2007 and Ohsumed datasets, while it has an acceptable result on the Genbase dataset, with respect to other MLL algorithms.

4.3 Discussion

The predictive performance of the PTMs varies according to dataset characteristics. Two types of multi-label datasets could be distinguished. The first type is the datasets with strong positive correlations among labels, such as the Yeast, Emotions, Flags and TMC2007 datasets. For this type, the correlation-based PTMs show a superior performance, using most of the evaluation metrics. The second type is the datasets with weak positive correlations among labels due to low cardinality (Scene and Genbase) or because the dataset does not have significant positive high-order correlations among labels, like Ohsumed dataset. For this type, the correlation-based PTMs show either a quite limited improvement or no improvement at all when compared against the frequency-based PTMs.

Among the correlation-based PTMs, the HSDF shows the best predictive performance and HAPCF and HAPCSDF nearly show an equal predictive performance. Among the frequency-based PTMs, the LFL transformation method shows a better predictive performance than the MFL transformation method.

In general, as the total number of the captured positive correlations among labels increases, the predictive performance of the MLR-PC becomes better. The only exception for this finding is the LFL, in which the total number of the captured positive correlations is the highest when using the LFL as a transformation method. Nevertheless, the predictive performance of the MLR-PC algorithm is not affected greatly by this large number of positive correlations due to the limited exploitation of these positive correlations because of the small frequency of the labels that exploited these positive correlations. Table 9 shows the total number of the captured positive correlations among labels with respect to the PTM being used.

Table 9. Total number of the captured positive correlations among labels with respect to the PTM being used.

Dataset	**HAPCF**	**HSDF**	**HAPCSDF**	**MFL**	**LFL**
Yeast	10	16	11	1	19
Emotions	4	4	4	1	4
Flags	8	7	8	3	9
Genbase	14	16	15	2	18
TMC2007	3	3	3	0	3

5. Conclusions and Future Work

In this paper, a new MLR algorithm called MLR-PC that captures positive correlations among class labels has been proposed. The captured positive correlations are exploited in the transformation step as well as in constructing a multi-label classifier. MLR-PC is a flexible algorithm, since any classifier could be used as a base classifier. Empirical analysis using different multi-label datasets, such as Yeast, Scene, Emotions and Flags, show that the MLR-PC algorithm is superior to other existing multi-label algorithms on several datasets, especially on those datasets with high cardinality.

High cardinality of a dataset is a strong evidence on the existence of significant correlations among labels and hence, MLR-PC algorithm showed a better performance with these datasets when utilizing correlation-based PTMs. Therefore, it is highly recommended to adopt a transformation criterion that considers the correlations among labels, especially with high-cardinality datasets, such as Yeast, Flags and TMC2007 datasets.

As for future work, more research that considers new PTMs based on the positive correlations among labels should be conducted. Also, capturing local and positive correlations among labels is a promising approach, especially in datasets with low cardinality, like the Genbase and Scene datasets.

References

[1] 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. 1-11, 2018.

[2] C. Zeng, W. Zhou, T. Li, L. Shwartz and G. Y. Grabarnik, "Knowledge Guided Hierarchical Multi-label Classification over Ticket Data," IEEE Transactions on Network and Service Management, vol. 14, no. 2, pp. 246-260, 2017.

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

[4] J. Huang, X. Qu, G. Li, F. Qin, X. Zheng and Q. Huang, "Multi-view Multi-label Learning with View-label-specific Features," IEEE Access, vol. 7, pp. 100979-100992, 2019.

[5] 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.

[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] 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.

[8] X. Kong, B. Cao and P. S. Yu, "Multi-label Classification by Mining Label and Instance Correlations from Heterogeneous Information Networks," Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614-622, [Online], Available: https://doi.org/10.1145/2487575.2487577, 2013.

[9] 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.

[10] M. Alluwaici, A. K. Junoh, F. K. Ahmad, M. F. M. Mohsen and R. Alazaidah, "Open Research Directions for Multi-label Learning," Proc. of IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 125-128, Penang, Malaysia, 2018.

[11] 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.

[12] 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.

[13] M. R. Boutell, J. Luo, X. Shen and C. M. Brown, "Learning Multi-label Scene Classification," Pattern Recognition, vol. 37, no. 9, pp. 1757-1771, 2004.

[14] 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, New York, NY, USA, 2009.

[15] S. Peters, L. Denoyer and P. Gallinari, "Iterative Annotation of Multi-relational Social Networks," Proc. of the IEEE International Conference on Advances in Social Networks Analysis and Mining, pp. 96-103, DOI: 10.1109/ASONAM.2010.13, Odense, Denmark, 2010.

[16] S. H. Ma, H. B. Le, B. H. Jia, Z. X. Wang, Z. W. Xiao, X. L. Cheng et al., "Peripheral Pulmonary Nodules: Relationship between Multi-slice Spiral CT Perfusion Imaging and Tumor Angiogenesis and VEGF Expression," BMC Cancer, vol. 8, no. 1, DOI: 10.1186/1471-2407-8-186, 2008.

[17] K. Trohidis, G. Tsoumakas, G. Kalliris and I. P. Vlahavas, "Multi-label Classification of Music into Emotions," In: J. P. Bellow, E. Chew and D. Turnbull (Eds.), Proceedings of the 9th International Conf. on Music Informative Retrieval, pp. 325-330, Philadelphia, PN: Drexel University, USA, 2008.

[18] 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.

[19] A. Elisseeff and J. Weston, "A Kernel Method for Multi-labelled Classification," Advances in Neural Information Processing Systems, Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01), pp. 681-387, 2001.

[20] A. Skabar, D. Wollersheim and T. Whitfort, "Multi-label Classification of Gene Function using MLPs," Proceedings of the 2006 IEEE International Joint Conference on Neural Network, pp. 2234-2240, DOI: 10.1109/IJCNN.2006.247019, Vancouver, BC, Canada, 2006.

[21] A. Chan and A. A. Freitas, "A New Ant Colony Algorithm for Multi-label Classification with Applications in Bioinfomatics," Proc. of the 8th Annual Conf. on Genetic and Evolutionary Computation (GECCO'06), pp. 27-34, [Online], Available: https://doi.org/10.1145/1143997.1144002, 2006.

[22] S. Diplaris, G. Tsoumakas, P. A. Mitkas and I. Vlahavas, "Protein Classification with Multiple Algorithms," Proc. of Panhellenic Conference on Informatics (PCI), pp. 448-456, Part of the LNCS, vol. 3746, Springer, Berlin, Heidelberg, 2005.

[23] K. Kawai and Y. Takahashi, "Identification of the Dual Action Antihypertensive Drugs Using TFS-based Support Vector Machines," Chem-Bio Informatics Journal, vol. 10, no. 9, pp. 41–51, Retrieved from: http://www.cbi.or.jp, 2010.

[24] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler and L. Schmidt-Thieme, "Multi-relational Matrix Factorization Using Bayesian Personalized Ranking for Social Network Data," Proceedings of the 5th ACM International Conference on Web Search and Data Mining, pp. 173-182, [Online], Available: https://doi.org/10.1145/2124295.2124317, 2012.

[25] L. Tang and H. Liu, "Relational Learning *via* Latent Social Dimensions," Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), pp. 817-826, [Online], Available: https://doi.org/10.1145/1557019.1557109, 2009.

[26] T. Li, C. Zhang and S. Zhu, "Empirical Studies on Multi-label Classification," Proc. of the 18th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'06), pp. 86-92, Arlington, VA, USA, 2006.

[27] O. A. Nassar and N. A. Al Saiyd, "The Integrating between Web Usage Mining and Data Mining Techniques," Proc. of the 5th IEEE International Conference on Computer Science and Information Technology, pp. 243-247, Amman, Jordan, 2013.

[28] J. Huang, Q. Feng, Z. Xiao, C. Zekai, Y. Zhixiang, Z. Weigang and H. Qingming, "Improving Multi-label Classification with Missing Labels by Learning Label-specific Features," Information Sciences, vol. 492, pp. 124-146, 2019.

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

[30] 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.

[31] S. S. Ibrahiem, S. S. Ismail, K. A. Bahnasy and M. M. Aref, "Convolutional Neural Network Multi-emotion Classifiers," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 2, pp. 97-108, Aug. 2019.

[32] R. Alazaidah, F. K. Ahmad and M. F. M. Mohsin, "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.

[33] M. L. Zhang and Z. H. Zhou, "ML-KNN: A Lazy Learning Approach to Multi-label Learning," Pattern Recognition, vol. 40, no. 7, pp. 2038-2048, 2007.

[34] J. Fürnkranz and E. Hüllermeier, "Pairwise Preference Learning and Ranking," Proc. of the European Conference on Machine Learning (ECML), pp. 145-156, LNCS, vol. 2837, Springer, Heidelberg, 2003.

[35] J. Fürnkranz, E. Hüllermeier, E. L. Mencía and K. Brinker, "Multilabel Classification *via* Calibrated Label Ranking," Machine Learning, vol. 73, no. 2, pp. 133-153, 2008.

[36] G. Tsoumakas, I. Katakis and I. Vlahavas, "Effective and Efficient Multi-label Classification in Domains with Large Number of Labels," Proc. of ECML/PKDD 2008 Workshop on Mining Multi-dimensional Data (MMD'08), vol. 21, pp. 53-59, 2008.

[37] J. Read, "A Pruned Problem Transformation Method for Multi-label Classification," Proc. of 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008), pp. 143-150, 2008.

[38] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An Ensemble Method for Multi-label Classification," Proc. of the European Conference on Machine Learning, pp. 406-417, LNCS, vol. 4701, Springer, Berlin, Heidelberg, 2007.

[39] M. Zhang and L. Wu, "Lift: Multi-label Learning with Label-specific Features," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 1, pp. 107-120, DOI: 10.1109/TPAMI.2014.2339815. 2015.

[40] L. Rokach, A. Schclar and E. Itach, "Ensemble Methods for Multi-label Classification," Expert Systems with Applications, vol. 41, no. 16, pp. 7507-7523, 2014.

[41] 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.

[42] E. Frank and H. Witten, "Generating Accurate Rule Sets without Global Optimization," Proceedings of the 15th International Conference on Machine Learning (ICML '98), pp. 144–151, 1998.

[43] T. Scheffer, "Finding Association Rules that Trade Support Optimally Against Confidence," Intelligent Data Analysis, vol. 9, no. 4, pp. 381-395, 2005.

[44] R. R. Bouckaert, "Properties of Bayesian Belief Network Learning Algorithms," Uncertainty Proceedings

1994, pp. 102-109, Morgan Kaufmann, [Online], Available: https://doi.org/10.1016/B978-1-55860-332-5.50018-3, 1994.

[45] W. W. Cohen, "Fast Effective Rule Induction," Proceedings of the 12th International Conference on Machine Learning, pp. 115-123, Tahoe City, California, 1995.

[46] M. Sumner, E. Frank and M. Hall, "Speeding up Logistic Model Tree Induction," Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases Discovery (PKDD'05), pp. 675-683, Heidelberg: Springer-Verlag, DOI: 10.1007/11564126_72, 2005.

[47] R. Kohavi, "The Power of Decision Tables" Proceedings of the 8th European Conference on Machina Learning (ECML'95), pp. 174-189, Berlin, Heidelberg: Springer-Verlag, DOI: 10.1007/3-540-59286-5_57, 1995.

[48] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update, " ACM SIGKDD Explorations Newsletter, vol. 11, no.1, pp. 10-18, 2009.

[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. Jul., pp. 2411-2414, 2011.

[50] S. Zhu, X. Ji, W. Xu and Y. Gong, "Multi-labelled Classification Using Maximum Entropy Method," Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05), pp. 274-281, [Online], available: https://doi.org/10.1145/1076034.

1076082 2005.

[51] 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.

[52] J. Huang, Z. Pingzhao, Z. Huiyi, L. Guorong and R. Haowei, "Multi-label Learning *via* Feature and Label Space Dimension Reduction," IEEE Access, vol. 8, pp. 20289-20303, 2020.

[53] F. Alshraiedeh, S. Hanna and R. Alazaidah, "An Approach to Extend WSDL-based Data Types Specification to Enhance Web Services Understandability," International Journal of Advanced Computer Science and Applications, vol. 6, no. 3, pp. 88-98, 2015.

ملخص البحث:

يعدّ تصنيف الأوسام المتعددة نوعاً عاماً من أنواع التصنيف والذي جذب اهتمام العديد من الباحثين في العقدين الماضيين، نظراً لقابليته للتطبيق في الكثير من المجالات الحديثة؛ مثل تصنيف المشاهد، وتكنولوجيا المعلومات الحيوية، وتصنيف النصوص، من بين مجالاتٍ أخرى. ويسمح هذا النوع من التصنيف بأن تكون الأمثلة مرتبطة بأكثر من وَسْم واحدٍ من أوسام الأصناف في آنٍ واحد. إنّ ترتيب أوسام الأصناف مسألة حاسمة في البحث المتعلق بتصنيف الأوسام المتعددة؛ نظراً لما له من تأثير مباشر في أداء المصنِّفات النهائية، لا سيما أنّ الأوسام التي تحتل المراتب العليا تحصل على فرصةٍ أعلى للتطبيق. تقدم هذه الورقة خوارزمية جديدة لترتيب الأوسام المتعددة، تسمى "ترتيب الأوسام المتعددة بناءً على الارتباطات الموجبة بين الأصناف (MLR-PC)". وتلتقط الخوارزمية المقترحة الارتباطات الموجبة بين الأوسام للتقليل من الحيّز الضخم للبحث وتحدد المرتبة الحقيقية لكل وسمٍ من أوسام الأصناف في المسائل المتعلقة بتصنيف الأوسام المتعددة.

والأهم من ذلك أنّ الخوارزمية المقترحة تستخدم طرقاً مبتكرة لتحويل المسائل، الأمر الذي يسهل استغلال الارتباطات الموجبة الدقيقة بين الأوسام. وهذا من شأنه أن يحسن الأداء التنبؤي لنماذج التصنيف المشتقة. وفد كشفت النتائج التجريبية باستخدام مجموعة بيانات متعددة الأوسام وخمسة من مقاييس التقييم أن الخوارزمية المقترحة (MLR-PC) تتفوق على الخوارزميات الأخرى شائعة الاستخدام في عمليات التصنيف.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

AN EFFICIENT HOLY QURAN RECITATION RECOGNIZER BASED ON SVM LEARNING MODEL

Khalid M.O. Nahar, Ra'ed M. Al-Khatib, Moy'awiah A. Al-Shannaq and Malek M. Barhoush

(Received: 29-Jun.-2020, Revised: 17-Aug.-2020, Accepted: 6-Sep.-2020)

ABSTRACT

Holy Quran recitation recognition refers to the process of identifying the type of recitation, among those authorized styles of recitation ("Qira'ah" in Arabic). Several previous studies investigated the recitation rules ("Ahkam Al-Tajweed" in Arabic) that are applied by readers or reciters while reading the Holy Quran aloud, but no study has examined the problem of tracking the type of recitation used in the reading. Through this research, we can assist Holy Quran students to easily learn the perfect and accurate recitation by successfully applying Ahkam Al-Tajweed and help them distinguish between different recitations or "Qira'ah". In this paper, a recognition model is conducted to recognize the "Qira'ah" from the corresponding Holy Quran acoustic wave. This model was built upon three phases; the first phase is the Mel-Frequency Cepstrum Coefficients (MFCC) feature extraction of the acoustic signal and labeling it, the second phase is training Support Vector Machine (SVM) learning model the labeled features and finally, recognizing "Qira'ah" based on this trained model. To attain this, we have built our corpus, which has 10 categories, each of which is labeled as one type of Holy Quran recitation or "Qira'ah". Different machine learning algorithms were applied and compared. Experimental results proved the superiority of our proposed SVM-based recognition model for "Qira'ah" over other machine learning algorithms with a success rate of 96%.

KEYWORDS

Arabic language, Quran recitation, Artificial Neural Network, Support Vector Machine (SVM).

1. INTRODUCTION

The key principle of communication is mainly to exchange ideas among peers and friends. People generally communicate and understand each other through speech. However, this might prove difficult for some people due to the wide variety of languages spoken globally. Nowadays, many computer applications in the area of computational linguistics have been designed to take into consideration the problem of recognizing and translating spoken language [1]. The productivity of such software applications statistically enhances and enriches the many disciplines that exist within the natural language processing field. In the last decade, computer scientists have paid special attention to developing efficient algorithms to recognize spoken words in the Speech Recognition (SR) domain. Progress in this domain has been significant and it is now widely known as Automatic Speech Recognition (ASR) technology [2]-[3].

ASR has been developed to recognize voices, which can improve communication among humans. These applications of ASR can compensate for the difficulties which are caused by the existence of such a wide range of languages in the world [4]. Therefore, several techniques have been introduced in the area of ASR [5]-[6]. The Support Vector Machine (SVM) is a powerful technique that has been widely used for speech recognition [7]-[8]. An ASR-based system recognizes spoken words by detecting and analyzing the input voice in a waveform, as illustrated in Figure 1 which shows the framework of ASR.

In this research, we employed a speech recognition model for recognizing "Qira'ah" within the readings from the Holy Quran. We focused on adapting and deploying the SVM algorithm with a new data corpus and special attributes as well as new rules. The proposed model initially converts the reader's sound waves (i.e., the proposed data corpus) into MFCC features and then a features vector matrix is generated. Parts of the extracted features are utilized to train the adapted SVM-based algorithm. The trained SVM developed was tested again with the other part of the extracted features and yielded very promising

K. M. O. Nahar, R. M. Al-Khatib, M. A. Al-Shannaq and M. M. Barhoush are with Computer Sciences Department, IT Faculty, Yarmouk University, Irbid, Jordan. Emails: khalids@yu.edu.jo, raed.m.alkhatib@yu.edu.jo, moyawiah.s@yu.edu.jo and malek@yu.edu.jo

results. As far as we know, this is the first time that SVM has been used in recognizing features of Holy Quran recitations. Our proposed SVM-based model was evaluated using real-world data collected from famous reciters of the Holy Quran. The experiments showed very fruitful outcomes when analyzing and comparing the obtained results. For comparative evaluation, the results obtained from our proposed SVM-based model were compared with other well-known algorithms, using the same datasets of waves collected. Interestingly, our proposed SVM-based model outperforms other algorithms in terms of accuracy in almost all experiments.

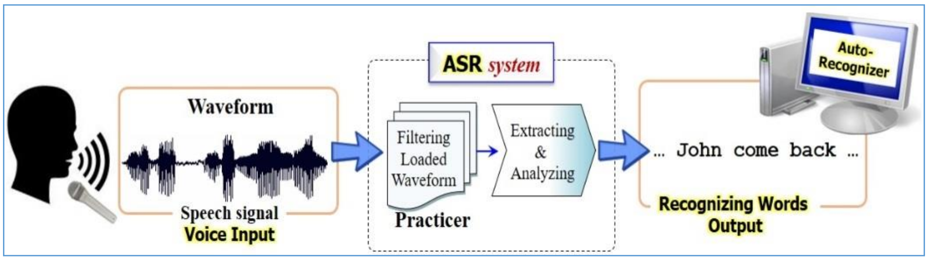


Figure 1. A general framework of Automatic Speech Recognition (ASR).

The rest of this paper is organized as follows. In Section 2, the literature review section, previous studies related to the current study are discussed and analyzed. In Section 3, we present the background of different speech recognition algorithms. The proposed approach and the methodological steps are fully presented in Section 4. Section 5 illustrates the experimental results. Finally, in Section 6, we show the conclusions of this study and propose several recommendations for further studies.

2. Literature Review

This section presents a basic description of a Speech Recognition System (SRS). It also provides more information about previous speech recognition systems used for Arabic.

2.1 Speech Recognition System (SRS)

Automatic Speech Recognition (ASR) enables the computer to identify the utterances of a person speaking into a microphone or telephone. Human Computer Interaction (HCI) is used, for example, as an authentication technique for user login *via* a voice recognition tool. Some ASR applications include voice interface as a command recognition application for computer users, dictation and written-text correction, interactive communication and voice response as an aid in learning foreign languages and for voice-controlled operation of machines. Additionally, ASR technology can improve the quality of life for disabled people, allowing them to communicate with others and interact in society [8]. In practice, the main principle of an ASR system is to recognize the appropriate word patterns for spoken utterances by applying a digitized analyzing process to enter analog sound waves [9]-[10].

2.2 Arabic Speech Recognition

One of the oldest Semitic languages in the world is the Arabic language. It is officially the sixth most spoken language and one of the official languages of the United Nations. There is an official Arabic linguistic form known as Modern Standard Arabic (MSA), which is mainly used in formal media, courtrooms, offices and by instructors for teaching in schools and universities [11]. Recently, many ASR systems have been developed in the domain of Arabic Speech Recognition to recognize the MSA version of Arabic. Unfortunately, recognizing traditional Arabic is still a challenge due to its lexical variety and the scarcity of data. Besides, the Arabic language is considered one of the most complex languages due to the morphological variations of its letters [12], [10].

ASR Arabic language research is still in its infancy age compared to the ASR already used in research related to other languages [13]-[14]. Consequently, we will review the top five studies that have been developed to improve Arabic speech recognition. In [15], the author dealt with continuous Arabic speech recognition, addressing the labeling of Arabic speech. Another model for Arabic speech recognition focused on prominent problems in recognizing conversational, dialectal and colloquial Arabic speech [13]. The authors reported significant improvement with respect to word error rate according to the 1997

NIST benchmark evaluations. An Arabic ASR system using ANN techniques was developed to improve the Arabic automatic recognition process [15]-[16]. Another Arabic ASR based on Hidden Markov Model (HMM), SVM or a hybrid of both was also developed [14], [17]. The last area relates to the work of some Arabic ASR researchers who took into consideration pronunciation variations to improve the performance of Arabic ASR systems [18].

2.3 Holy Quran Recitation Recognizers

Computer-aided Pronunciation Learning (CAPL) was considered early in the twentieth century when great efforts were made and improvements achieved by researchers. Recognizing the Holy Quran Tajweed rules and tracking reading errors presented a great challenge [19]. The authors in [20], developed an intelligent Tajweed-rules tracker system. The system listens to the Holy Quran, recited by a learner and then suggests a correction to his/her recitation. The name of the system they developed is “Hafize”. The “Hafize” system is trained to recognize the recitations of 10 reciters, including women, men and children. The accuracy of “Hafize” recitation was measured against each reader’s recitation with the average result in the region of 89%. The “Hafize” system has a limitation in that it relates to a pronunciation based on phonetic rules which are not given. “Hafize” does not only consider the verse as a whole, but it can also recognize mistakes at word level. The authors in [21], developed another recitation model to help Malaysian primary school students pronounce the Quran verses correctly. Unfortunately, we have not found sufficient information about the implementation of this model to be able to test it. A new recitation system named “Makhraj” was developed by [22] to make the recitation of the Holy Quran less dependent on expert reciters. The accuracy of this model was calculated based on the False Rejection Rate (FRR). The authors used MFCC for feature extraction. Two modes for recognition process are used in the Makhraj-based system: the one-to-one mode and the one-to-many mode. However, the one-to-one mode is 98% accurate, which is not considered very accurate in this mode due to the utilization of a simple matching technique.

The authors in [23] introduced an SVM-based learning model, which recognizes Quranic words from online resources. In [24], the researchers implemented a virtual learning system called “Electronic Miqra’ah”. The system can independently receive voice commands and allows blind students to interact with the voice of Holy Quran scholars. However, the system has a low recognition rate due to the use of the Google API. Nonetheless, as a model, it works well. In [25], the authors developed a high-performance phoneme-based and speaker-independent system for Holly Quran recitation based on HMM with 3-emitting states. The accuracy of this system reaches 92%. In [26], the authors introduced a powerful training system for Quran recitation using a continuous Arabic speech recognition system that depends on HMM to recognize Quran recitation and detect recitation errors. Their Arabic speech recognition system provides phonetic time alignments and its classifier is used to distinguish between confusing phones. This Arabic speech recognition system was built on the results produced by WEKA tool. In this system, The SVM classifier was used with an accuracy rate of 91.2% at word level.

In [27], the authors used the CMU-Sphinx4 tool to produce a new recitation recognition system for the Holy Quran based on an HMM algorithm. In this HMM-based recognition system, a simplified set of phonemes was used to build the language model, as well as to train the recognizer. The authors in [28] developed a system for tracking Holy Quran basic Tajweed rules using deep learning techniques. They used MFCC, WPD and HMM-SPL feature extraction algorithms and considered them as the best features. Moreover, they reported that their system reached a 97.7% rate of accuracy. A limited number of Quranic chapters were tested and the accuracy was high, reaching 98%. Table 1 summarizes the studies that have been presented in this section.

Table 1. Summary of Holy Quran recitation research.

AUTHORS & DATE	**SYSTEM DESCRIPTION & REF.S**	**Strengths**	**Weakness**	**Feature Extraction Algorithm**
Muhammad et al. (2012)	Intelligent Tajweed-rules tracker system (Hafize) [20]	It can discover mistakes in the recitation of verses from the Holy Quran.	It did not work at the phonetic level.	MFCC

Mssraty & Faryadi (2012)	A recitation model to help Malaysian primary school students [21]	An initial analysis denotes its potential usefulness.	It has not yet been implemented.	N/A
Arshad et al. (2013)	A recitation system: “Makhraj” [22]	It produces a good level of accuracy on one-to-one mode.	Accuracy level is low in the one-to-many mode due to the simple match technique used.	MFCC
Sabbah & Selamat (2014)	A learning model based on SVM to recognize Quranic words [23]	Good recognition accuracy.	The sparsity of the feature matrix; and the number of features increases the time for building the classification model.	Statistical Features
Mohamed et al. (2014)	A virtual learning system: (Electronic Miqra’ah) [24]	Robust application.	Moderate recognition rate due to the use of Google API.	N/A
Elhadj et al. (2014)	Phoneme-based speaker-independent system for Holy Quran recitation [25]	Good accuracy level reaching 92%.	N/A	MFCC
Tabbaa & Soudan (2015)	Quran recitation based on HMM and SVM classifier [26]	Accuracy reached 91.2% at the word level.	It suffers from high confusing sounds.	MFCC
El Amrani et al. (2016)	Limited Holy Quran recitation based on HMM model [27]	Accuracy reaches 98%, but limited corpus.	Not all Arabic phonemes are included in the recognition.	MFCC
Al-Ayyoub, Damer & Hmeidi (2018)	Verifying the proper use of Tajweed-rules of Holy Quran [28]	Accuracy of the system reached 97.7%.	It only tackled the basic Tajweed rules without using correct recitation verification.	MFCC, WPD & HMM-SPL

Despite the examples given in the literature review, only limited attempts have been made in Quranic recitation recognition to date. Thus, the automatic identification and recognition of aspects of a Quran recitation is still a fresh field for study. In this paper, a new Quranic recitation recognition model is proposed based on the SVM learning algorithm, which is directed at recognizing and detecting aspects of Holy Quran recitation using what we consider to be an enhanced approach.

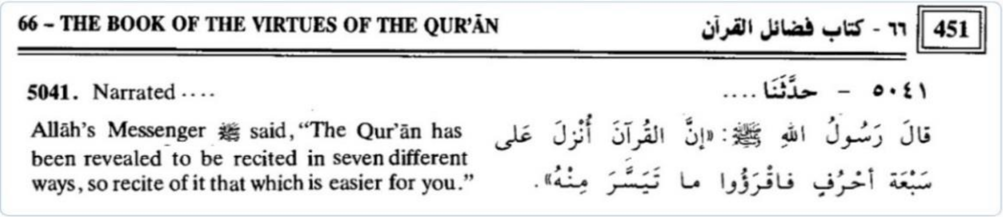
3. BACKGROUND

In this section, the types of Holy Quran reciters are discussed. This section also presents an overview of the most outstanding learning algorithms.

3.1 Types of Holy Quran Reciters Based on Narration “Qira’ah”

In various Arab communities, there are many different regional accents,“lahjah” “لهجة” in Arabic. These numerous accents are seen as a challenge for the ASR-based applications, which need to recognize the correct accent among multiple local accents. It is noticeable that some of these applications are unfamiliar with local accents [30]-[31]. Arabic is one of the most prevalent languages in the world, because of its number of speakers and because the Holy Quran was given through divine revelation in the Arabic language [32]. As a result, not only do the Arab communities in 23 Arabic countries speak Arabic, but also all non-Arabic speakers who share this faith aim at learning Arabic to recite the Holy Quran correctly [33]-[34]. Their main goal is to understand and read the Holy Quran properly, based on the officially established rules of readings [35]. Unlike any other subject of voice or speaker identification systems, the way the reader recites Holy Quran is different from other types of speech. The difference resides in the existence of certain acoustic rules, which are known as "Ahkam Al-Tajweed" ("أحكام التجويد"), previously mentioned, that have to be applied and maintained when reading the Holy Quran. Furthermore, emotional features are added by the reader. Moreover, the transition from one acoustic level to another when reading also distinguishes this type of reading or recitation from normal speech. The transition from one acoustic level to another in the Arabic language is known as "Maqam" "مقام" [19].

The Holy Quran has seven main designated styles of reading or "Qira'at" "قراءات", which are accepted as the most popular ways of reading or recitation extracts from the Holy book. This is what is stated by Prophet Mohammed in the 4th century AH, according to the narrative "hadith" "حديث" No. 5041, extracted from [36], as shown in Figure 2 [37].



66 – THE BOOK OF THE VIRTUES OF THE QUR'ĀN ٦٦ - كتاب فضائل القرآن 451

5041. Narrated .... ٥٠٤١ - حدَّثنا ....

Allāh's Messenger ﷺ said, "The Qur'ān has been revealed to be recited in seven different ways, so recite of it that which is easier for you."

قالَ رَسُولُ اللهِ ﷺ: «إنَّ القُرآنَ أُنْزِلَ عَلى
سَبْعَةِ أحْرُفٍ فاقْرَؤُوا ما تَيَسَّرَ مِنْهُ».

Figure 2. Extracted from the Hadith 5041 (the rendering of the quotation by Sahih Al-Bukhari).

Subsequently, three new reading styles were added to the seven "Qira'at". The first seven of the accepted readings are known in the Arabic language as "Mutilator Qira'at", meaning "successive readings", in Arabic "قراءات متواترة" [38]. The last three readings are called "Mashhur Qira'at" "قراءات مشهورة". Here, the term "Qira'at" "قراءات" relates to the different recitations that represent changes in Holy Quran reading styles that occur mainly in the areas of pronunciation and tone uttering Quranic extracts. In the science of reading of the "Qira'at", the ten styles of reading or recitation are attributed to the original readers known as "Imam" "إمام" and designated by the name of the Imam. The most popular "Qira'at" variants are the "Asim" "قراءة عاصم" and the "Nafi" reading styles "قراءة نافع". All recitations are listed in Table 2. The differences in the recitation styles can be limited to three main distinctions:

a) The prolongation "Mudud" "مدود" and the shortening of words called "Kasser" "قصر".
b) Adding the punctuation (in Arabic "Harakatt" "حركات") of the written text of the Holy Quran.
c) The pronunciation and reading of the Quranic extracts and parts, depending on the varying styles of recitation [21][8].

Many researchers have addressed the various challenges people meet in approaching the Arabic language, in addition to the intricacies of the Holy Quran and the way they should recite its verses. One of the main drivers of this work is to facilitate the recitation by non-Arabic speakers [21][39]. It is recognized that beginners face many difficulties in reciting, distinctly for the following three reasons:

a) They lack oral practice and monitoring for correction of errors.
b) Many learners are unfamiliar with Arabic, but the revealed text is in the Arabic language.
c) Most software related to Tajweed lacks any follow-up for readers to improve in their future recitation.

As previously stated, some of the difficulties readers find in the reciting aloud of the Holy Quran are the result of the need to base their reading on strict rules of recitation "Ahkam Al-Tajweed" [35][10]. The science of "Tajweed" teaches the reciter the basic rules that help him/her to pronounce the words of the Holy Quran as Prophet Muhammad, peace be upon him, recited them. A teacher of Tajweed must be authorized and certified to do so.

Since the teaching of the correct pronunciation of the words and the appropriate rendering of Quranic verses is central to Islam, nowadays we have the opportunity to check the accuracy of the recitation automatically using applications related to ASR systems as noted earlier. Recitation from the Holy Quran is not like any other type of reading. The rules of recitation ("Ahkam Al-Tajweed") must be followed to be a faithful rendering of the verses. Because these rules of pronunciation are followed by multiple speakers, distinguishing between different speakers is not easy and becomes an important issue for researchers. However, no study has covered the area comprehensively until now.

Different approaches in the literature use MFCC features along with some learning algorithms, such as Hidden Markov Model (HMM), Support Vector Machine (SVM) and Artificial Neural Network (ANN), in an attempt to recognize a voice among a set of different voices [14], [17], [27]-[28] . A recent investigation was conducted to build a recognizer to identify the different types of Holy Quran reading “Qira’ah” [37]. However, the authors built a limited corpus of only two types of reading styles: “Eldori” “قراءة الدوري” and “Hafs” “قراءة حفص”. In addition to that, the research depends on one reciter, who recites using the two styles of recitation in the study. Another limitation of the study is that only three chapters (“Surahs” “السور”) were included in their corpus. We believe that these limitations may not produce objective results.

3.2 Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is inspired by the human biological nerve system [40]. ANN is a collection of many artificial neurons that are connected together for the purpose of learning. The main purpose of ANN is to map the inputs into meaningful outputs. To illustrate this further, ANN could be found in many topologies, such as Feed Forward and Back Propagation. Figure 3 shows the architecture of a feed-forward ANN.

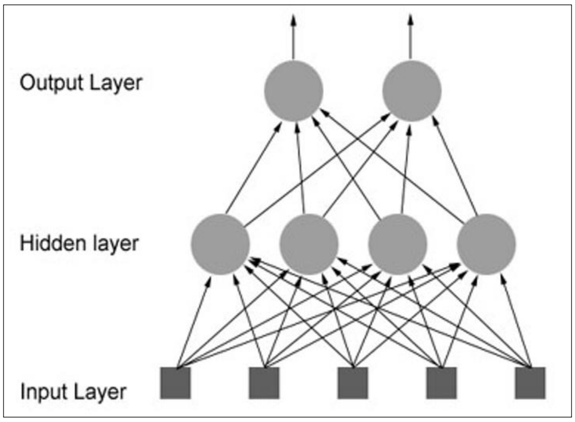


Figure 3. Feed-forward artificial neural network.

Each input from the input layer feeds each node in the hidden layer and then the hidden layer feeds the next layer until it reaches the output layer. In most cases, ANN consists of multiple hidden layers that must be passed through before ultimately reaching the output layer. The ANN in Figure 3 is called a feed-forward ANN, due to the fact that signals are passed through the layers of the neural network in a single forward direction. However, the ANN can be feedback networks, where the architecture allows signals to travel in both directions [41].

3.3 Support Vector Machine (SVM)

The Decision Tree, Radial Bases, Forest Decision Trees, Nearest Neighbor, Fuzzy Classifier, Deep Learning Classifier and Support Vector Machine are some of the well-known machine learning algorithms used in the literature for classification and categorization [40], [42], [46]. The most significant of these algorithms is SVM, which is considered the best of learning algorithms [42].

SVM is a common classifier which separates instances through the use of a hyperplane. In supervised learning, SVM produces an optimal straight line that separates categories, as shown in Figure 4. In essence, the SVM algorithm finds a decision boundary with maximum margins between categories, because the optimal separating hyperplane maximizes the margin of the training data [43]. The SVM is a built-in function in a variety of software. The Sequential Minimal Optimization (SMO) algorithm is an SVM implementation of WEKA open source software, which was implemented by [44]-[45]. The SMO is one of the most efficient solutions for the SVM algorithm. It is based on solving a series of small quadratic problems, which in each iteration uses only two variables in the working set in order to save time [46].

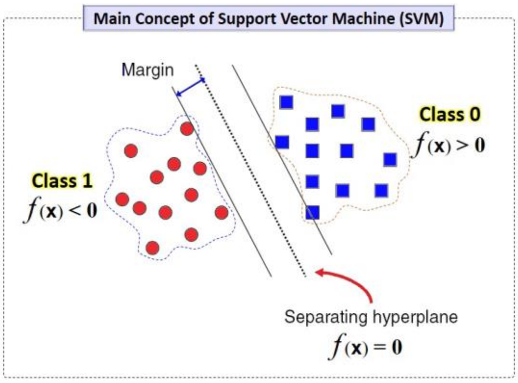


Figure 4. SVM classification hyperplane.

The SVM as a classifier is considered to be one of the most powerful statistical learning techniques [46][42]. SVM successfully addresses and solves different kinds of problems, for three key reasons. The first reason is that the tangent of SVM which is seen in its mathematical and theoretical foundations supports classification and problem-solving. The second reason is that SVM has been confirmed as being suitable to manage complex data, including data of high dimensions, such as text and image data [40][14]. The third one is that SVM has the potential for success in the pattern recognition domain and its effectiveness has already been confirmed in the image processing field [47]. In brief, the SVM consists of class classifiers, which are mathematically constructed from the summation of a kernel function as stated in the following equation:

$$f(x) = \sum_{i=1}^{N} \alpha_i t_i K(x, x_i) + d \,, \tag{1}$$

where $K(x, x_i)$ constructs the summation of the kernel function modeled by [48]. t_i denotes the ideal outputs when $\sum_{i=1}^{N} \alpha_n t_i = 0$ and $\alpha_i > 0$. From the training set of optimization process in [49], x_i denotes the support vectors. The ideal and optimal results may be 1 if the corresponding support vector is in class 0 or may be -1 if the corresponding support vector is in class 1. In terms of classification, the class decision happens when the value of $f(x)$ is above a specific threshold or below it. The kernel function $K(.,.)$ is constrained and limited to specific properties, which are known as the Mercer condition and can be expressed as follows:

$$K(x, y) = b(x)^t . b(y) \,, \tag{2}$$

where $b(x)^t$ implies a mapping according to the input space. Here, x indicates what could possibly be an infinite dimensional space. Finally, the Mercer condition here is responsible for guaranteeing that the validation of the margin concept and the optimization of the SVM limited to definite and particular boundaries [50].

Specifically, the optimization condition depends on a maximum margin concept, as depicted in Figure 4. The SVM chooses an appropriate high-dimensional space to put in place the best hyperplane that has the maximum margin. As a result, the training of the input data points set will be located on the boundaries of the support vectors that are based on Equation (1). These boundaries are represented by two solid lines, as shown in Figure 4. Modeling of these two boundaries is the main aim of the SVM training process.

3.4 Hidden Markov Model (HMM)

The HMM model makes a chain called a Markov chain usually used in stochastic processing [51]. The HMM has the capability to heuristically address the variability using such stochastic modeling. Furthermore, HMM model was efficiently used to improve the behavioral performance of metaheuristics [52]. Since the HMM is a time series learning algorithm, it is not adapted to the comparative outputs to our problem and consequently not used as a model in this research.

In this paper, after due consideration, it was decided to use the SVM learning algorithm, because SVM generates the hyperplane that classifies the training instances with high speed and more accuracy than other traditional clustering methods. Those other traditional clustering methods mainly depend on probability distributions when training data is classified as has been demonstrated by [53].

4. Proposed SVM-based Approach

The proposed approach is mainly based on the feature extraction process and recitation modeling. Figure 5 illustrates the proposed reading/recitation recognition system, where some parts of this figure are adapted from [54]. Initially, the proposed system is trained and tested using SVM, ANN, among others. After building the corpus of recitation of the Holy Quran, the proposed recognizer is built by applying three phases. The first one is to use the Mel-Frequency Cepstrum Coefficients (MFCCs) algorithm to extract a range of informative media features from the trained corpus. The second one is to formulate a matrix of training features and build a learning model across the SVM learning algorithm. Lastly, testing and comparing the results obtained from the previous phase to show the superiority and relevance of the SVM algorithm to the problem of Quran recitation recognition. In the testing phase, some external data (data totally outside the corpus) will be used to test and evaluate our proposed SVM-based approach. The following subsections show more details about our methodology.

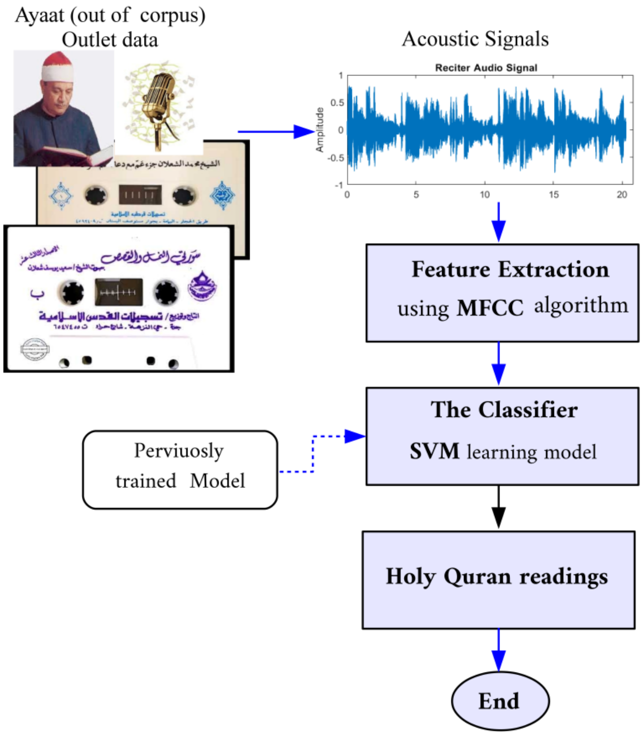


Figure 5. Overview of proposed SVM-based approach.

4.1 Building Corpus

Since the proposed approach aims to identify Holy Quran reading types, which are well-known as “Qira’ah” “قراءة”, acoustic samples from Holy Quran need to be collected in order to build a relevant corpus. The corpus contains a number of acoustic waves that are labeled based on the “Qira’ah”. A pictorial view of an acoustic wave and its features are illustrated in Figure 6 and in Figure 7, respectively. Acoustic waves for each reading were collected and the feature vectors of these acoustic waves were extracted using the MFCC extraction algorithm. Figure 6 represents a sample of the acoustic wave used in building up the corpus along with its full energy spectrum. Each acoustic wave has its own phase (starting wave angle), amplitude and frequency, but all of them are limited within a specific range and duration.

Figure 7 represents a sample of the MFCC features and below is the spectrum of the acoustic wave after removing weak energy. Almost all spectra of all waves are normally distributed. In the spectrum, each feature has a value which represents part of the wave attributes.

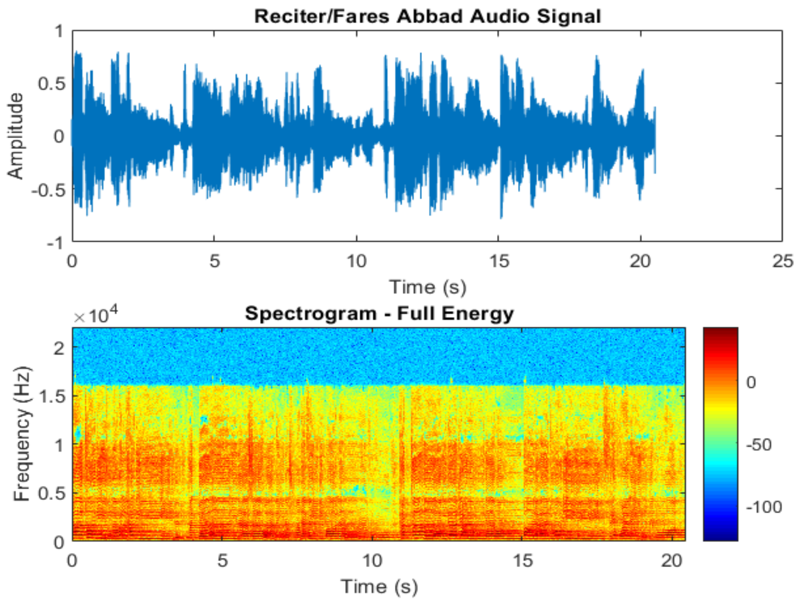


Figure 6. Sample of acoustic wave with its full spectrum [54].

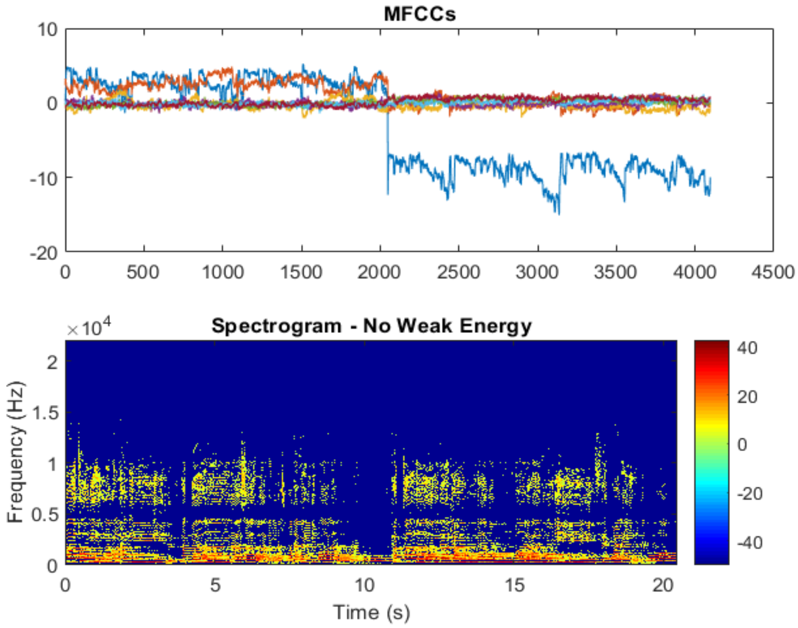


Figure 7. Sample of MFCC file without weak energy spectrum [54].

4.2 Orders and Categories for Labeling Reciters

This sub-section introduces details of the data acquisition and corpus construction related to our proposed method. In order to build a recitation recognition system for Holy Quran "Qira'ah", the main features of the "Qira'ah" waves should be isolated from those of other waves. In brief, the Holy Quran is a collection of verbal revelations given to Prophet Muhammad over a period of twenty-three years. The Holy Quran has 114 Chapters ("Surah" in Arabic "سورة") of varying lengths and each Chapter "سورة" consists of a number of individual verses "Ayaat" "آيات". There are 6,348 different verses in the Holy Quran. To date, there is no corpus for Holy Quran verses "Ayaat" "آيات" based on the style of reading or recitation or "Qira'ah" "قراءة". Therefore, in this research, we have collected all possible "Qira'ah" waves and placed them in a folder, regardless of the identity of the reader. Ten different folders have been created to represent ten different types of reading ("Qira'ah") waves. Table 1 shows four columns. In the first, the name of the Imam who established the specific "Qira'ah" is mentioned in

Arabic. In the second column, the English translation of the imam name is given and it is abbreviated in the third column. In the last column, the number of waves that were collected with respect to each "Qira'ah" is given. It should be mentioned here that the collected wave files are available and can be downloaded from the official website of our proposed method.

Ten different types of reading ("Qira'ah") were used in the construction of the corpus, as detailed in Table 2. Each Holy Quran reading type has a different number of wave files. As a result, we have a total of 258 wave files. Holy Quran readers often used the same styles of reading or recitation and rarely use the other available styles. Therefore, some types of reading are represented by a large number of sound waves, while others do not.

Table 2. Reading types, code and number of audio files.

The Imam of the "Qira'ah"	Translated title in English	Key-Class	Number of Wave Files
ابن عامر	Ebin-Amer	EA	6
ابن كثير	Ebin-Khatheer	EK	6
ابو جعفر المدني	Abee-Jafar-Almadani	AJ	18
ابو عمرو البصري	Abee-Amro-Albasree	AA	51
الكسائي	Al-Kesaee	K	51
حمزة الكوفي	Hamzah-Alkofee	HK	18
خلف العاشر	Khalaf-AlAsher	KH	3
عاصم الكوفي	Aseem-Alkofee	AK	48
نافع المدني	Nafee-AlMadani	NM	51
يعقوب الحضرمي	Yakoob-Alkhathramee	YKH	6
Ten "Qira'ah"	Total		258 files

4.3 Extracting Features and Building the Feature Vectors

Many previous studies have been carried out in converting sound waves, including the extraction of statistical information from acoustic signals. These studies have resulted in many valuable methods for interpreting the wave to provide information that could be processed easily. A comparative study between those methods was performed by [55]. The methods that were examined in Shrawankar's study include: Linear Predictive Coefficients (LPCs), Linear Predictive Cepestral Coefficients (LPCCs), Perceptual Linear Predictive Coefficients (PLPs), Mel-frequency Cepestral Coefficients (MFCCs), Mel

Table 3. Ranking of feature extraction methods.

Feature Extraction Function	Best Classification Algorithm	Accuracy
MFCC	SVM	89.16%
LPC	Random Tree	79.15%
LPCC	Random Tree	76.99%
LAR	Decision Table	83.83 %
SSC	Function, Logistic	66.05%
LSF	Random Tree	82.54%
PLP	Decision Tree	71.23%
FFT	ANN	78.34%
MEL	SVM	87.61%
RASTA	ANN	85.23%
DELTA	Random Forest and SVM	82.50%

Scale Cepestral Analysis (MEL), Relative Spectra Filtering of Log Domain Coefficients (RASTA), First Order Derivative (DELTA), Perceptual Linear Prediction (PLP), Fast Fourier Transform (FFT), Line Spectral Frequencies (LSFs), Spectral Subband, Centroid (SSC) and Log Area Ratio Coefficients (LARs) [55]. Finally, a comprehensive analysis and wide investigation proved that the MFCC method produces a small number of coefficients that represent frequency information [55]. It replicates the perception of loudness in the human auditory system and is considered as a simplified auditory model that eases computation and is relatively faster than others.

Before proceeding to our experimental part, we undertook a brief investigation regarding the accuracy that could be gained from each type of feature extraction mechanism. Table 3 shows the best results of this combination based on a brief initial test. From Table 3, it is clear that MFCC method produced the maximum accuracy when used with SVM learning algorithm. A pictorial view of Table 3 is shown in Figure 8.

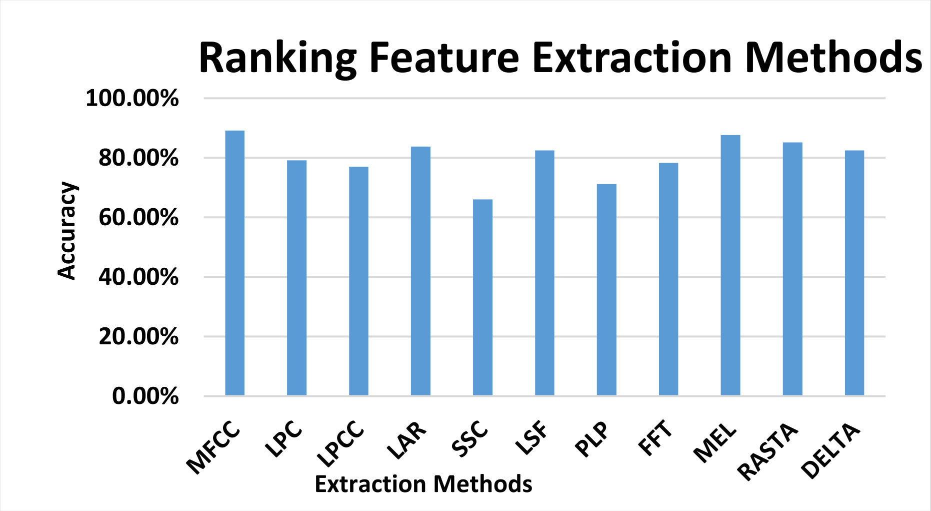
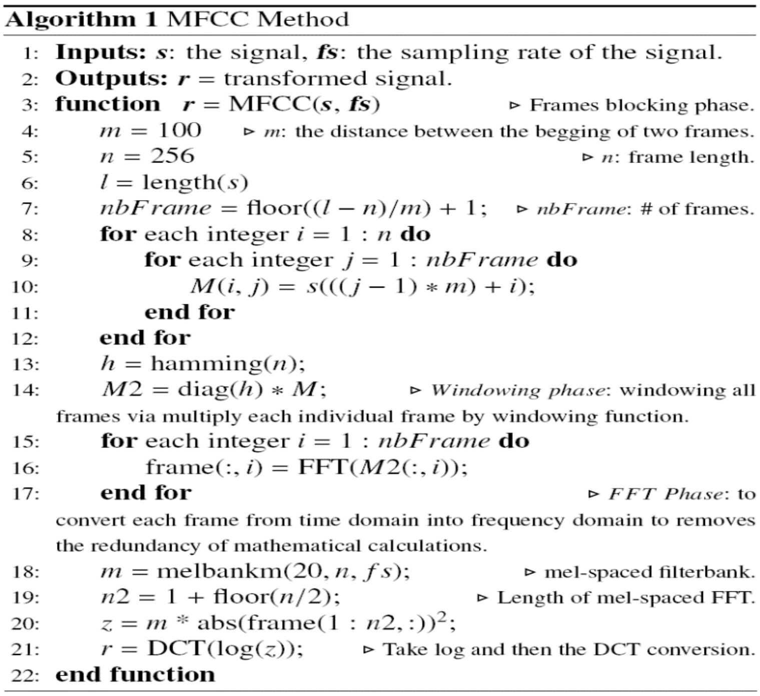


Figure 8. Feature extraction methods and levels of accuracy.



```
Algorithm 1 MFCC Method
Inputs: $s$: the signal, $fs$: the sampling rate of the signal.
Outputs: $r$ = transformed signal.
function $r = \text{MFCC}(s, fs)$                ▷ Frames blocking phase.
    $m = 100$      ▷ $m$: the distance between the begging of two frames.
    $n = 256$                                     ▷ $n$: frame length.
    $l = \text{length}(s)$
    $nbFrame = \text{floor}((l - n)/m) + 1$;      ▷ $nbFrame$: # of frames.
    for each integer $i = 1 : n$ do
        for each integer $j = 1 : nbFrame$ do
            $M(i, j) = s(((j - 1) * m) + i)$;
        end for
    end for
    $h = \text{hamming}(n)$;
    $M2 = \text{diag}(h) * M$;          ▷ Windowing phase: windowing all
  frames via multiply each individual frame by windowing function.
    for each integer $i = 1 : nbFrame$ do
        $\text{frame}(:, i) = \text{FFT}(M2(:, i))$;
    end for                                       ▷ FFT Phase: to
  convert each frame from time domain into frequency domain to removes
  the redundancy of mathematical calculations.
    $m = \text{melbankm}(20, n, fs)$;             ▷ mel-spaced filterbank.
    $n2 = 1 + \text{floor}(n/2)$;             ▷ Length of mel-spaced FFT.
    $z = m * \text{abs}(\text{frame}(1 : n2, :))^2$;
    $r = \text{DCT}(\log(z))$;      ▷ Take log and then the DCT conversion.
end function
```

After using the MFCC, the speech signal was divided into segments of 15 ms frames with the use of a hamming window for further analysis [56]. We experimentally determined that 15 ms frames generated better recognition than others, because Holy Quran readers generally have a slow rate of recitation. A larger frame size does not enhance the ability of the recognition system to learn the characteristics of the signal. Feature vectors were extracted by MFCC algorithm as illustrated in Algorithm 1.

MFCC is a short-period power spectrum that is used to represent sound waves [57]. Mel frequencies are based on the critical bandwidth of the human ear recognized as a variation with frequency filters, which includes two types of frequencies [58]; the first one at frequencies below 1 kHz and the second one logarithmic filters at frequencies higher than 1 kHz to capture phonetically important characteristics [59]. In recitations from the Holy Quran, the correct pronunciation depends on the context, controlled by the voice of the reader and the reader's ability to move from one acoustic level to another "Maqam" "مقام". The stages involved in MFCC extraction are illustrated in Figure 9.

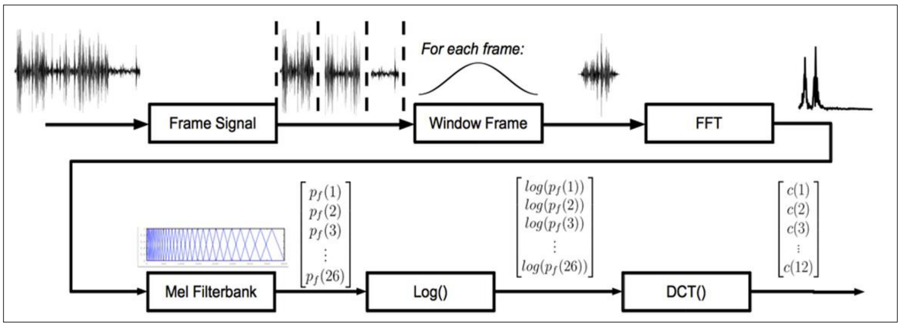

For each frame:
Frame Signal
Window Frame
FFT
Mel Filterbank
Log()
DCT()
$p_f(1)$ $p_f(2)$ $p_f(3)$ ⋮ $p_f(26)$
$log(p_f(1))$ $log(p_f(2))$ $log(p_f(3))$ ⋮ $log(p_f(26))$
$c(1)$ $c(2)$ $c(3)$ ⋮ $c(12)$

Figure 9. MFCC feature extraction stages [54].

The MFCC is calculated using Equation (3) and its implementation is in Algorithm 1.

$$C_i = \sum_{k=1}^{N} X_k cos\left(\frac{[\pi_i(k-0.5)]}{N}\right), \quad for\ i = 1,2,\dots,p \tag{3}$$

where C_i denotes the Cepstral coefficients, p is the order, k is the number of discrete Fourier transformations magnitude coefficients, X_k is the k^{th} order log-energy output of the filter bank and N is the number of filters (usually 20). Thus, 19 coefficients and an energy feature were extracted, generating a vector of 20 coefficients perframe. In this research, the first 20 orders of the MFCC were extracted. It was proven by Chaudhari (2015) that the increasing number of filters would raise the recognition rate, though the recognition and computational time for both training and testing would be negatively affected by the increasing number of filters [38].

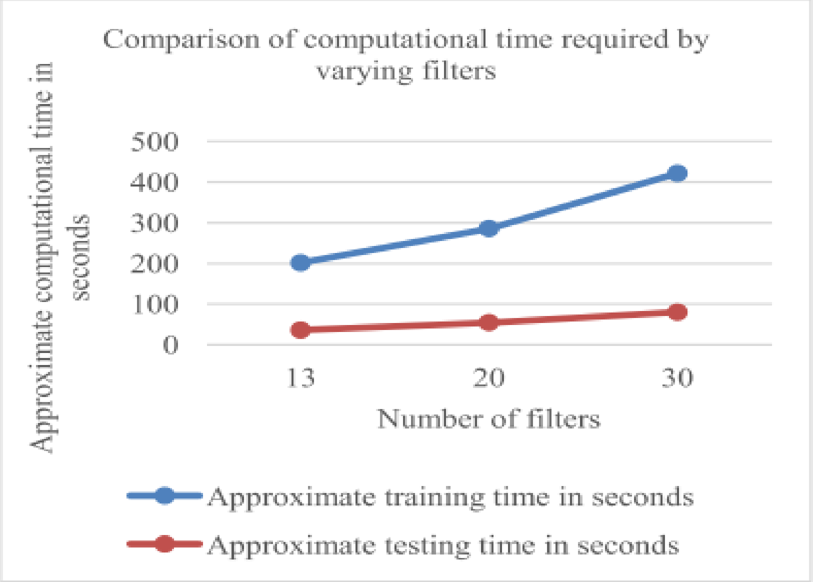


Figure 10. Comparison of computational time required by varying filters (taken from [38]).

It can be concluded from Figure 10 and Chaudhari (2015) that the recognition rate will be decreased either for testing or training as the number of filters increased and *vice versa* [38]. In our paper, 20 filters will be used, the number used in most speech recognition tools, in accordance with the belief that this number of filters provides a better rate of recognition and an acceptable computation time for training and testing.

Since the verses are of different lengths, 20 feature representations of each verse ("Ayah" "آية") were extracted for the 258 acoustic waves. The MFCC files correspond to the acoustic waves, extracting the feature vectors of the total number of verses in a vector matrix for all readers with a size equal to (20×22952) features. After that, these 20-feature matrices were combined into one file for the ten different styles of recitation. Each verse vector is transposed and labeled according to the reading style in one CSV-file that includes a labeled matrix for reading. Furthermore, we have generated six feature vector matrices using six different methods for the purpose of comparison between the levels of accuracy of the results of different feature extraction methods.

4.4 Formulating the Training Matrix and Its Feature Representation Using SVM

The CSV labeled matrix extracted and generated in the previous phase is divided into two parts. Part I usually takes 70% of this matrix which is used for training on the proposed model. The rest of the matrix (30%) is used to test and evaluate the proposed model. Figure 11 shows the general framework of the training phase, while Figure 12 shows the general testing framework.

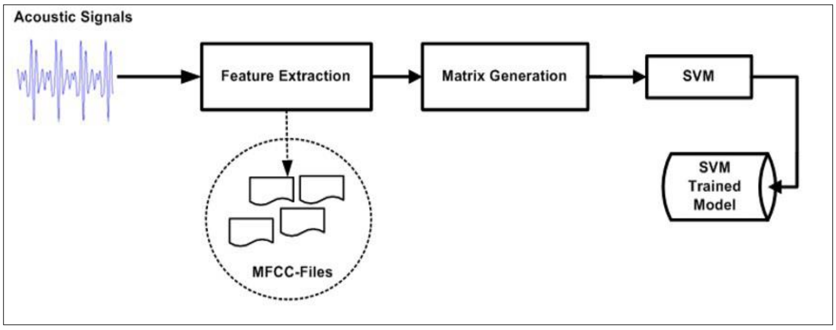


Figure 11. Proposed SVM-based training model.

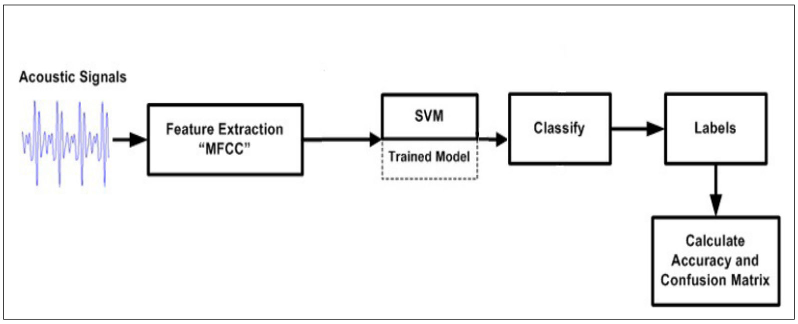

Acoustic Signals
Feature Extraction "MFCC"
SVM
Trained Model
Classify
Labels
Calculate Accuracy and Confusion Matrix

Figure 12. Proposed testing SVM-based model.

The selection of the features was homogeneous and normally distributed. Figure 13 shows the homogeneity in selecting the features drawn by the WEKA tool. Features one and two for all the readings were biased a little to the right, while the other features were normally distributed in all the readings of the audio waves. This is due to the limited examples of some types of reading, leading to a limited number of features. Figure 14 is the same as Figure 13, but expressed in a graphical representation using colored dots.

A built-in randomized filter is used to rearrange the features in randomly chosen rows. This randomized filter will enhance the learning process for any learning algorithm and will affect the accuracy of the results.

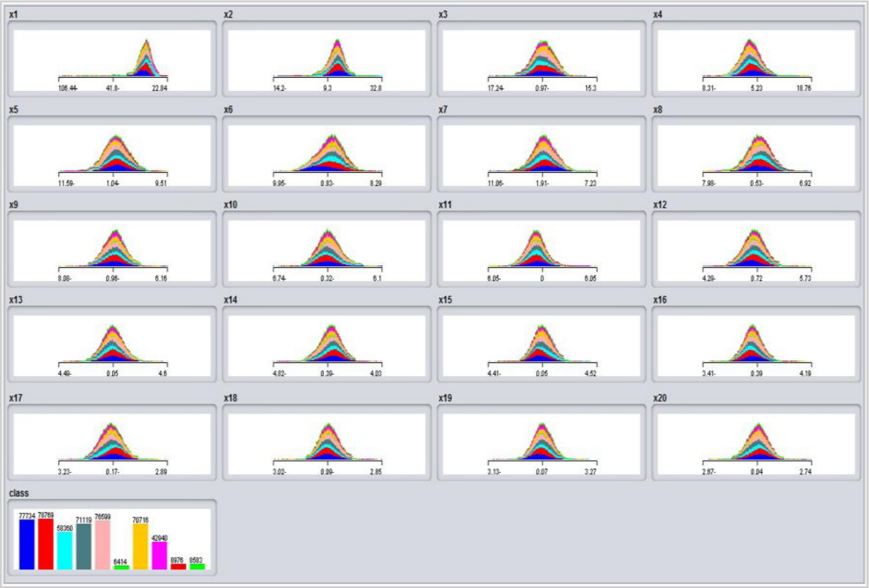


Figure 13. Feature distribution over audio wave files for readings.

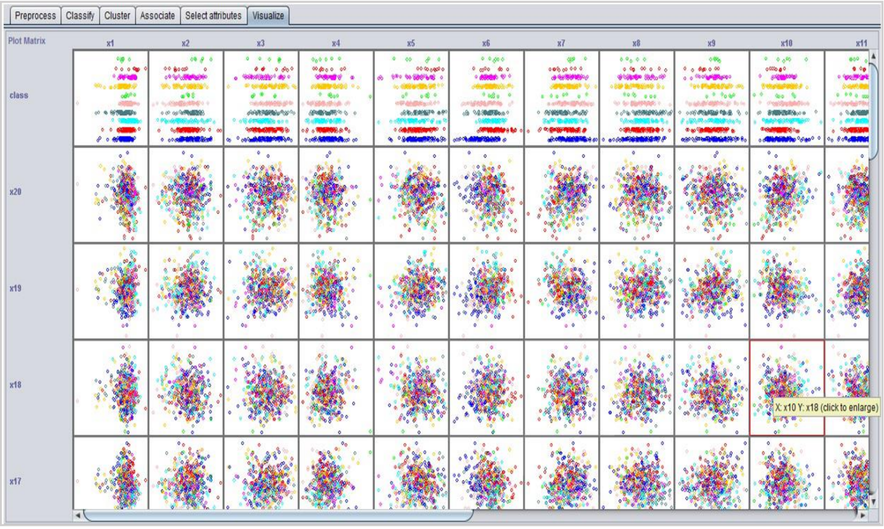


Figure 14. Representation of feature homogeneity.

4.5 Testing Phase

After obtaining the training model through SVM learning, the rest of the feature matrix (30%) is used for testing, as shown in Figure 12. The testing results obtained are represented by the confusion matrix of SVM, which shows the hit and the misclassification of the waves. Note that the SVM confusion matrix will be fully discussed in Section five. In addition, the proposed methodology stages will be applied to different learning algorithms other than the SVM. The results obtained from these learning algorithms will be compared for analysis and discussion.

5. Experimental Results and Discussion

The Holy Quran recitation recognition system is based primarily on the characteristics extracted from acoustic waves in conjunction with learning and classification algorithms. The SVM was trained with 10 Holy Quran readings, as listed in Table 1. For comparison purposes, the same experiment was repeated with the Least Square Support Vector Machine (LSSVM) and ANN models. For training purposes, we chose 70% of the features to train SVM, LSSVM, ANN and other learning algorithms. The remaining 30% of the features were cropped for testing purposes. When the attribute matrix is loaded into WEKA, attribute classifications and properties (features) from X_1 to X_{20} are displayed, so that the user can verify them. The attributes $X_1, X_2, \ldots, X_{20}$ are the feature vector values or predictors and the "Readings" are the target classes. We started training SVM on 70% of the features and testing on 30%.

The output of the SVM text file contains the testing confusion matrix, accuracy matrix for all classes and summary of the overall results. The confusion matrix is a data structure used to show the classification results. The rows of this matrix represent the desired classification, while the columns represent the predicted classification. Table 4 represents the confusion matrix of the testing phase with SVM. It is obvious that some samples go below the hyper plane of the SVM (main diagonal), while others are above it. In either case, it must be regarded as a misclassification of the sample. When the samples are on the main diagonal, they are correctly classified. From Table 4, most of the samples are correctly classified.

Table 4. Testing confusion matrix of SVM.

a	b	c	d	e	f	G	h	i	j	<-- classified as		
22239	338	102	231	162	5	181	132	14	28	**a**	=	HK
81	**23169**	51	175	82	9	117	33	13	4	**b**	=	AJ
116	160	**16752**	113	42	0	74	41	6	7	**c**	=	AK
206	177	68	**20713**	98	23	126	30	17	6	**d**	=	AA
221	115	35	131	**22383**	1	121	29	12	18	**e**	=	K
36	66	10	70	18	**1679**	40	3	30	11	**f**	=	YKH
180	121	52	173	82	1	**20411**	18	2	12	**g**	=	NM
80	96	36	57	34	0	33	**12310**	24	10	**h**	=	KH
50	140	40	99	58	28	22	40	**2292**	0	**i**	=	IA
46	18	12	50	42	12	74	31	6	**2301**	**j**	=	IK

To calculate the accuracy from the confusion matrix, Equation (4) is used:

$$\text{Accuracy} = \frac{\sum \text{Diagonal Sample of confusion matrix}}{\text{Total Sample}} \qquad (4)$$

The total accuracy of the recitation's recognition is 96.59% when using SVM. About 4% of the total samples were poorly classified. Misclassified samples may arise as a result of the way in which the Holy Quran reader recites different verses. The similarity between the reader's sound and its emotional tone is critical. Most of Holy Quran readers follow the same rules for the recitation, Ahkam Al-Tajweed "أحكام التجويد", when they are reciting the various verses of the "surah" "السور". Applying "Ahkam Al-Tajweed" appropriately will increase the chance of the similarity of the acoustic waves produced by different readers. Finally, every reader has his/her own shifts in tone "Makam" "مقام", but these might sound very similar when the rules are applied. The detailed levels of accuracy of SVM at the class are clearly established in Table 5. The weighted average for each measurement is listed in the last row. As

Table 5. Detailed SVM accuracy by class.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.949	0.008	0.956	0.949	0.953	0.944	0.971	0.916	HK
0.976	0.01	0.95	0.976	0.963	0.956	0.983	0.931	AJ
0.968	0.003	0.976	0.968	0.972	0.968	0.982	0.949	AK
0.965	0.009	0.95	0.965	0.957	0.95	0.978	0.921	AA
0.97	0.005	0.973	0.97	0.972	0.967	0.983	0.949	K
0.855	0.001	0.955	0.855	0.902	0.903	0.927	0.819	YKH
0.97	0.006	0.963	0.97	0.966	0.961	0.982	0.938	NM
0.971	0.003	0.972	0.971	0.971	0.969	0.984	0.946	KH
0.828	0.001	0.949	0.828	0.884	0.884	0.913	0.788	IA
0.888	0.001	0.96	0.888	0.922	0.922	0.944	0.854	IK
0.961	**0.006**	**0.961**	**0.961**	**0.961**	**0.955**	**0.978**	**0.929**	**←Weighted Average**

can be seen from Table 5, the weighted average of precision reached 96%, while the weighted average of the false positive rate (FP) reached 0.006 which is a very good indicator regarding the recognition rate.

The TP, FP, Precision, Recall, F-Measure, MCC, ROC and PRC measurements are mentioned in Table 5, while the details of these measurements are reported in Table 6, which is taken from the URL[1]. Note that the weighted arithmetic mean is similar to an ordinary arithmetic mean (the most common type of average), except that instead of each of the data points contributing equally to the final average, some data points contribute more than others, see the (Taken from [2]).

Table 6. Summary of measurements and accuracy metrics.

Measurements	Meaning
TP	Rate of true positives (instances correctly classified as a given class).
FP	Rate of false positives (instances falsely classified as a given class).
Precision	Proportion of instances that are truly of a class divided by the total instances classified as that class
Recall	Proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP rate).
F-Measure	A combined measure for precision and recall calculated as: $F - Measure = \frac{2*Precision*Recall}{Precision+Recall}$. It is the weighted harmonic mean (sometimes called the subcontrary mean), which is one of several kinds of averages appropriate for situations when the average of rates is desired.
MCC	The Matthews Correlation Coefficient whichis used in machine learning as a measure of the quality of binary (two-class) classifications.
ROC Area	Receiver Operating Characteristic: A plot of a true positive fraction (= sensitivity) *vs.* a false positive fraction (= 1 – specificity) for all potential cut-offs for a test.
PRC Area	Precision-recall curve: A plot of precision (= PPV) *vs.* recall (= sensitivity) for all potential cut-offs for a test. PRC might be a better choice for unbalanced datasets.

The pictorial view of the previous table is represented in Figure 15, which illustrates the graphical representation of SVM results. It is clear that all classes of readings are recognized with a high probability of accuracy. K and AK classes reach the highest recognition rate, which is mapped to “Aseem” “عاصم” and “Al-Kesaee” “الكسائي” readings. Table 7 summarizes the SVM results and measurements of error. Clearly, the absolute error rate is very low which is considered satisfactory in this kind of recognition problem. The overall accuracy as seen from Table 7 reaches 96%.

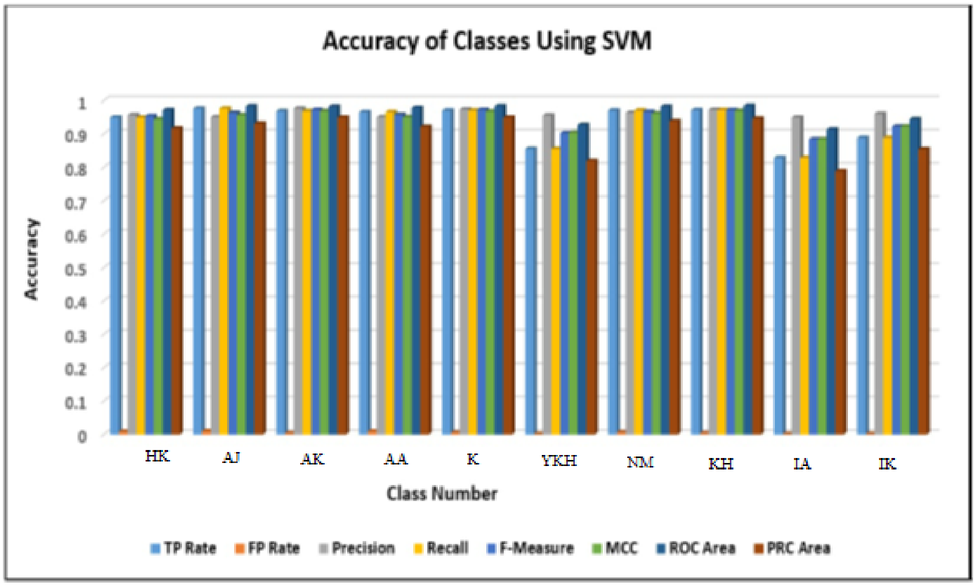


Figure 15. Graphical representation of SVM results.

[1] https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used
[2] https://en.wikipedia.org/wiki/Weighted_arithmetic_mean

Table 7. Summary of SVM results.

Correctly Classified Instances	144249 or 96.1256 %
Incorrectly Classified Instances	5814 or 3.8744 %
Kappa Statistic	0.9552
Mean Absolute Error	0.0077
Root Mean Squared Error	0.088
Relative Absolute Error	4.4758 %
Root Relative Squared Error	29.9198 %
Total Number of Instances	150063

The same experiment was repeated using multi-perceptron ANN with 20 inputs representing the features. Table 8 shows the confusion matrix of the ANN. Compared to Table 7, it is obvious that most of the instances are incorrectly classified.

Table 8. Testing confusion matrix of ANN.

a	b	c	d	e	f	G	h	i	j	<-- classified as		
15121	1356	751	1441	2028	2	1594	1120	4	15	**a**	=	HK
856	**17269**	658	1902	1007	3	1067	904	25	43	**b**	=	AJ
410	743	**13287**	801	526	2	477	993	33	39	**c**	=	AK
1312	790	1513	**14955**	1105	3	1364	371	41	10	**d**	=	AA
1417	552	541	876	**17671**	28	1425	532	5	19	**e**	=	K
136	341	156	656	185	**120**	197	155	17	0	**f**	=	YKH
942	734	1041	1694	1256	26	**14838**	518	3	0	**g**	=	NM
777	359	624	950	255	3	365	**9322**	9	16	**h**	=	KH
64	622	243	601	285	2	157	302	**493**	0	**i**	=	IA
203	52	275	220	511	0	357	439	0	**535**	**j**	=	IK

Moreover, by looking at Table 9, we can see that the maximum classification accuracy is 78%. Some classes, such as IA and IK, reached only 22.7% and 38.9%, respectively, which are very low compared to their accuracy when SVM is used. The average accuracy for all classes reaches only 69%, which indicates that the ANN is inadequate for this kind of recognition. This signifies that the absolute error was 8% when using ANN, while it was 0.0077 when using SVM. This result indicates that SVM is more appropriate than the ANN algorithm.

Table 9. Detailed ANN accuracy by class.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.645	0.048	0.712	0.645	0.677	0.622	0.896	0.712	HK
0.728	0.044	0.757	0.728	0.742	0.695	0.932	0.78	AJ
0.768	0.044	0.696	0.768	0.73	0.694	0.94	0.754	AK
0.697	0.071	0.621	0.697	0.656	0.597	0.909	0.649	AA
0.766	0.056	0.712	0.766	0.738	0.689	0.916	0.772	K
0.061	0	0.635	0.061	0.112	0.194	0.56	0.118	YKH
0.705	0.054	0.679	0.705	0.692	0.641	0.916	0.701	NM
0.735	0.039	0.636	0.735	0.682	0.652	0.898	0.719	KH
0.178	0.001	0.783	0.178	0.29	0.369	0.782	0.227	IA
0.206	0.001	0.79	0.206	0.327	0.399	0.861	0.389	IK
0.69	**0.049**	**0.695**	**0.69**	**0.682**	**0.639**	**0.908**	**0.704**	**←Weighted Average**

A summary of ANN results and error measurements is shown in Table 10. Clearly, the absolute error rate is very high compared to SVM. The overall accuracy according to Table 10 reaches 69%.

For comparison between ANN and SVM, Table 11 shows a summary of the recognition rate between ANN and SVM when applied on testing data from the same corpus and data from outside the corpus (outlet data).

Our proposed SVM-based system obtained better results compared with ANN, having higher accuracy and lower Mean Square Error (MSE) compared with ANN. A stacked bar graph is shown in Figure 16 to illustrate the results for both SVM and ANN with different measurements.

Table 10. Summary of ANN results.

Correctly Classified Instances	103611 or 69.045 %
Incorrectly Classified Instances	46452 or 30.955 %
Kappa Statistic	0.6409
Mean Absolute Error	0.0831
Root Mean Squared Error	0.2155
Relative Absolute Error	47.9745 %
Root Relative Squared Error	73.2585 %
Total Number of Instances	150063

Table 11. SVM and ANN result summary.

Accuracy	**Learning Algorithms**	
	SVM	**ANN**
Accuracy-Normal Corpus	**96 %**	69%
Accuracy-Outlet Data	**95%**	60%
Average Accuracy	**95.5%**	64.5%
Average MSE	0.0077	0.08

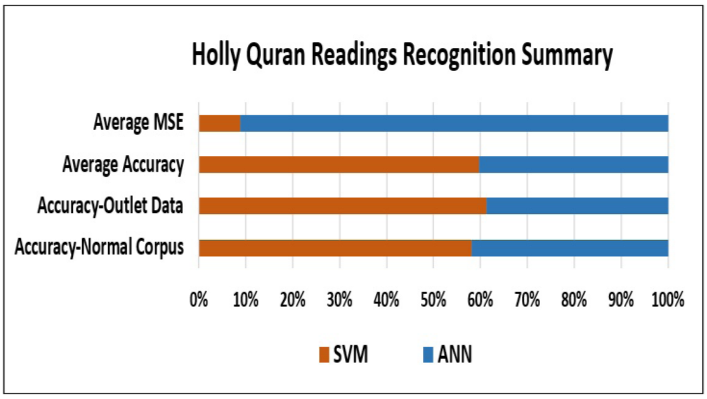


Figure 16. SVM and ANN result summary.

Table 12. Summary of results obtained from different learning algorithms.

Number of Instances = 500211 (70% for Training and 30% for Testing)				
Classifier	**Training time in seconds**	**Testing time in seconds**	**Mean Square Error (MSE)**	**Accuracy**
NB-Tree (Decision Tree)	72657.88	8261.66	0.2641	47.4118%
RBF (Radial Bases)	3211.61	2270.68	0.2796	31%
Random Forest (Forest Decision Trees)	1500.33	300.65	0.2293	68.3529%
NNge (Nearest Neighbor)	1200.25	300.21	0.3147	50.4706%
Multi-Objective Evolutionary Fuzzy Classifier	5000.66	1700.12	0.3163	20.8235%
Deep Learning Classifier	1900.33	460.89	0.2827	40.7647%
SVM (Support Vector Machine)	**42357.75**	**5211.73**	**0.08800**	**96.12 %**
LSSVM (Least Squares Support Vector Machines)	**32133.60**	**3332.00**	**0.09100**	**95.16 %**
Multi-layer Perceptron (ANN)	2677.02	10.20	0.2435	62.0588%

For the sake of consistency, further experiments were conducted using other learning algorithms, such as: NB-Tree (Decision Tree), RBF (Radial Bases), Random Forest (RF), NNge (Nearest Neighbor), Multi-Objective Evolutionary Fuzzy Classifier (MOEFC), Deep Learning Classifier (DLC) and Least Square Support Vector Machine (LSSVM). We focused on the training time, test time, MSE and accuracy measurements. The results obtained from those algorithms are shown in Table 12.

It is clear that SVM and LSSVM have the lowest MSE and the highest accuracy, but they need more time for training and testing compared to other algorithms. Figure 17 shows a chart of Table 12.

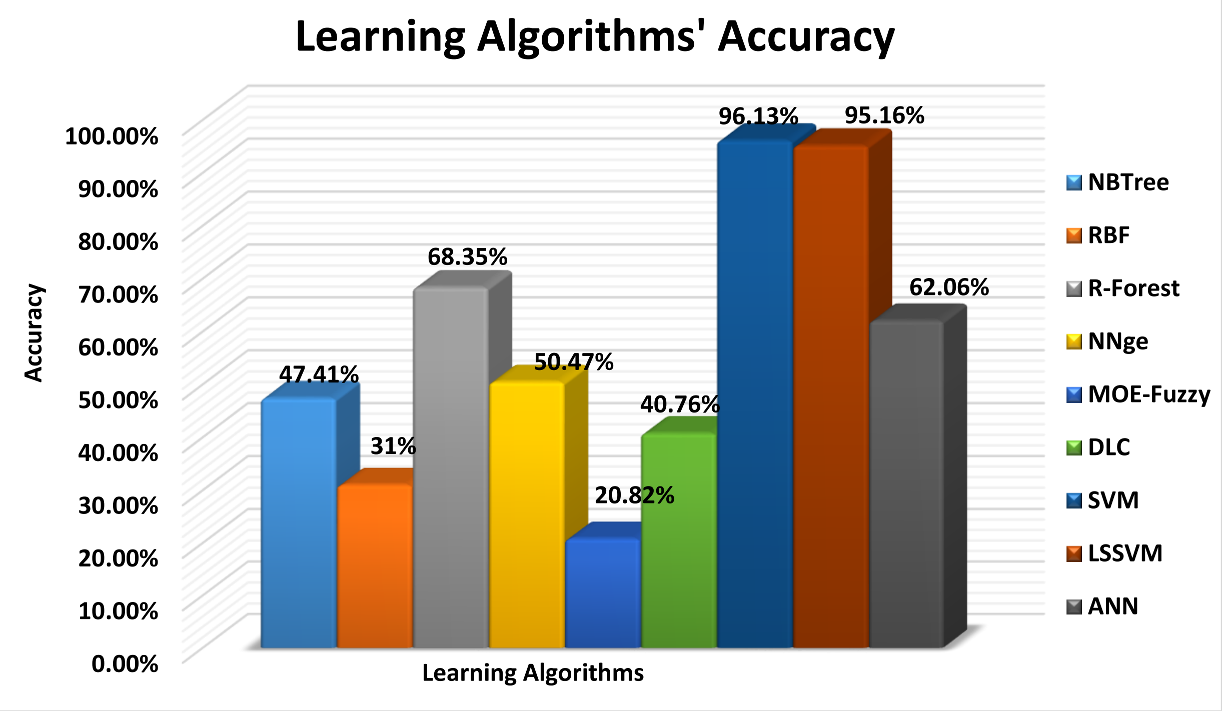


Figure 17. Accuracy of different learning algorithms.

6. Conclusion and Future Work

In this research, we have addressed the problem of determining the efficiency of recitation of the Holy Quran based on established rules of recitation. We have investigated the recognition of the type of Holy Quran recitation based on the SVM learning algorithm. Moreover, we compared the results with other learning algorithms. Acoustic waves for the ten Holy Quran reading styles ("Qira'ah" "قراءة") were collected in a corpus of ten recitation variants "Qira'at" "قراءات" and a variety of readers were included in the study. Subsequently, MFCC properties were extracted from wave signals and labeled according to the appropriate reading ("Qira'ah") style. The labeled matrix of the "Qira'ah" was fed to SVM using the WEKA tool. 70% of the classified matrix was given to SVM for training. The remaining 30% of the labeled matrix was used for testing purposes. Additional outlet data was used to test the proposed model in order to demonstrate the validity and reliability of our proposed system.

A comparison between our adopted learning algorithm (SVM) and other learning algorithms (Table 11) was made to validate the adequacy of SVM. Briefly, the results obtained using SVM outperformed the results obtained when using other learning algorithms. The results reveal that the identification accuracy using SVM is higher in comparison to those produced by the ANN and other learning algorithms. The accuracy using SVM reaches approximately 96%, while ANN accuracy is 62%. Some other learning algorithms, such as FDT, reached 68%, which is greater than ANN's level of accuracy. When comparing the results obtained from Table 11, we can see that SVM demonstrated its superiority over all other learning algorithms used. However, SVM requires more time in the training phase, while proving to be faster in testing input data.

The promising outcomes of this research have encouraged us to undertake further investigation in recognizing Holy Quran recitation. Several future directions could be taken in this area. For example, new deep learning algorithms could be applied to compare the results obtained from the application of such algorithms with the results of this study. Another example is hybridizing some powerful learning algorithms, like Learning Vector Quantization (LVQ) with the SVM model to improve the results. Furthermore, expanding the corpus to include other reciters could be carried out to validate our system. Finally, we may need to extend the evaluation criteria for more accurate measurement of SVM performance.

REFERENCES

[1] Y. Kato, "Speech Recognition System," Journal of the Acoustical Society of America, vol. 107, no. 5, p. 2326, 2000.

[2] L. R. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, PTR Prentice Hall Englewood Cliffs, 1993.

[3] T. K. Das and K. M. O. Nahar, "A Voice Identification System Using Hidden Markov Model," Indian Journal of Science and Technology, vol. 9, no. 4, DOI: 10.17485/ijst/2016/v9i4/83894, 2016.

[4] J. M. Baker et al., "Developments and Directions in Speech Recognition and Understanding, Part 1 [DSP Education]," IEEE Signal Processing Magazine, vol. 26, no. 3, pp. 75-80, 2009.

[5] J. Chong, E. Gonina, D. Kolossa, S. Zeiler and K. Keutzer, "An Automatic Speech Recognition Application Framework for Highly Parallel Implementations on the GPU," Technical Report No. UCB/EECS-2012-47, [Online], available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-47.html, 2012.

[6] K. Nahar, H. Al-Muhtaseb, W. Al-Khatib, M. Elshafei and M. Alghamdi, "Arabic Phonemes Transcription Using Data Driven Approach," Int. Arab J. of Information Technology, vol. 12, no. 3, pp. 237–245, 2015.

[7] Y.-H. Shao and N.-Y. Deng, "A Coordinate Descent Margin Based-twin Support Vector Machine for Classification," Neural Networks, vol. 25, pp. 114–121, 2012.

[8] K. Nahar, M. Abu Shquier, W. G. Al-Khatib, H. Al-Muhtaseb and M. Elshafei, "Arabic Phonemes Recognition Using Hybrid LVQ/HMM Model for Continuous Speech Recognition," International Journal of Speech Technology, vol. 19, no. 3, pp. 495–508, 2016.

[9] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent, "An Architecture for a Generic Dialogue Shell," Natural Language Engineering, vol. 6, no. 3–4, pp. 213–228, 2000.

[10] K. M. O. Nahar, W. G. Al-Khatib, M. Elshafei, H. Al-Muhtaseb and M. M. Alghamdi, "Data-driven Arabic Phoneme Recognition Using Varying Number of HMM States," Proc. of the 1st IEE International Conference on Communications, Signal Processing and Their Applications (ICCSPA 2013), pp. 1–6, Sharjah, United Arab Emirates, 2013.

[11] M. Abdul-Mageed, M. Diab and S. Kübler, "SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media," Computer Speech and Language, vol. 28, no. 1, pp. 20–37, 2014.

[12] F. Diehl, M. J. F. Gales, M. Tomalin and P. C. Woodland, "Morphological Decomposition in Arabic ASR Systems," Computer Speech and Language, vol. 26, no. 4, pp. 229–243, 2012.

[13] K. Kirchhoff et al., "Novel Approaches to Arabic Speech Recognition: Report from the 2002 Johns-Hopkins Summer Workshop," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03), vol. 1, pp. I344-I347, Hong Kong, China, 2003.

[14] E. Zarrouk, Y. Ben Ayed and F. Gargouri, "Hybrid Continuous Speech Recognition Systems by HMM, MLP and SVM: A Comparative Study," International Journal of Speech Technology, vol. 17, no. 3, pp. 223–233, 2014.

[15] A. Al-Otaibi, "Speech Processing," Britsh Library in Association with UMI, 1988.

[16] M. M. El Choubassi, H. E. El Khoury, C. E. J. Alagha, J. A. Skaf and M. A. Al-Alaoui, "Arabic Speech Recognition Using Recurrent Neural Networks," Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795) (ISSPIT 2003), pp. 543–547, Darmstadt, Germany, 2003.

[17] A. Ali et al., "Automatic Dialect Detection in Arabic Broadcast Speech," arXiv Paper, arXiv1509.06928, [Online], available: https://arxiv.org/pdf/1509.06928.pdf, 2015.

[18] F. Biadsy, N. Habash and J. Hirschberg, "Improving the Arabic Pronunciation Dictionary for Phone and

Word Recognition with Linguistically-based Pronunciation Rules," Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 397–405, United States, 2009.

[19] N. J. Ibrahim, M. Y. Zulkifli and H. Mohd, "Improved Design for Automated Tajweed Checking Rules Engine of Quranic Verses Recitation : A Review," International Journal on Quranic Research, no. January, pp. 39–50, 2011.

[20] A. Muhammad, Z. Ul Qayyum, M. Waqar Mirza, S. Tanveer, A. M. Martinez-Enriquez and A. Z. Syed, "E-hafiz: Intelligent System to Help Muslims in Recitation and Memorization of Quran," Life Science Journal, vol. 9, no. 1, pp. 534–541, 2012.

[21] T. Mssraty and Q. Faryadi, "Teaching the Qur'anic Recitation with Harakatt: A Multimedia-based Interactive Learning Method," International Journal of Scientific and Enginerring Research, vol. 3, no. 8, pp. 1–4, 2012.

[22] N. W. Arshad et al., "Makhraj Recognition for Al-Quran Recitation Using MFCC," International Journal of Intelligent Information Processing, vol. 4, no. 2, pp. 45–53, Jun. 2013.

[23] T. Sabbah and A. Selamat, "Support Vector Machine-based Approach for Quranic Words Detection in Online Textual Content," Proc. of the 8th Malaysian Software Engineering Conference (MySEC 2014), pp. 325–330, Langkawi, Malaysia, 2014.

[24] S. A. E. Mohamed, A. S. Hassanin and M. T. Ben Othman, "Virtual Learning System (Miqra'ah) for Quran Recitations for Sighted and Blind Students," Journal of Software Engineering and Applications, vol. 07, no. 04, pp. 195–205, 2014.

[25] Y. O. M. Elhadj, M. Alghamdi and M. Alkanhal, "Phoneme-based Recognizer to Assist Reading the Holy Quran," Recent Advances in Intelligent Informatics, Part of the Advances in Intelligent Systems and Computing Book Series (AISC), vol. 235, pp. 141–152, 2014.

[26] H. M. A. Tabbaa and B. Soudan, "Computer-aided Training for Quranic Recitation," Procedia-Social and Behavioral Sciences, vol. 192, pp. 778–787, 2015.

[27] M. Y. El Amrani, M. M. H. Rahman, M. R. Wahiddin and A. Shah, "Building CMU Sphinx Language Model for the Holy Quran Using Simplified Arabic Phonemes," Egyptian Informatics Journal, vol. 17, no. 3, pp. 305–314, 2016.

[28] M. Al-Ayyoub, N. A. Damer and I. Hmeidi, "Using Deep Learning for Automatically Determining Correct Application of Basic Quranic Recitation Rules," International Arab Journal of Information Technology, vol. 15, no. 3, pp. 620–625, 2018.

[29] M. Y. El Amrani, M. R. Wahiddin, M. M. H. Rahman and A. Shah, "Towards Using CMU Sphinx Tools for the Holy Quran Recitation Verification," International Journal on Islamic Applications in Computer Science and Technol., vol. 4, no. 2, pp. 10–15, 2016.

[30] A. Hughes, P. Trudgill and D. Watt, English Accents and Dialects: An Introduction to Social and Regional Varieties of English in the British Isles, Routledge, 2013.

[31] O. F. Zaidan and C. Callison-Burch, "Arabic Dialect Identification," Computational Linguistics, vol. 40, no. 1, pp. 171–202, 2014.

[32] H. K. Tayyeh, M. S. Mahdi and A. S. A. Al-Jumaili, "Novel Steganography Scheme Using Arabic Text Features in Holy Quran," International Journal of Electrical and Computer Engineering, vol. 9, no. 3, pp. 1910-1918, 2019.

[33] L. C. Moore, "Learning by Heart in Qur'anic and Public Schools in Northern Cameroon," Social Analysis, vol. 50, no. 3, pp. 109–126, 2006.

[34] A. Rasmussen, Women, the Recited Qur'an and Islamic Music in Indonesia, Univ. of California Press, 2010.

[35] Z. A. Adhoni, H. Al Hamad, A. A. Siddiqi, M. Parvez and Z. A. Adhoni, "Cloud-based Online Portal and Mobile Friendly Application for the Holy Qur'an," Life Science Journal, vol. 10, no. 12, 2013.

[36] M. M. Khan, The Translation of the Meanings of Sahih Al-Bukhâri, vol. 5, Kazi Publications, 1997.

[37] T. Wahbi and A. Gadeed, "The Recognition of Holy Quran Reading Types 'Rewaih," International Journal of Advanced Research in Computer Science, vol. 5, no. 3, pp. 37-40, 2014.

[38] A. A. Chaudhari, "Effect of Varying MFCC Filters for Speaker Recognition," International Journal of Computer applications, vol. 128, no. 14, pp. 7–9, 2015.

[39] M. N. Al-Kabi, G. Kanaan, R. Al-Shalabi, M. O. K. Nahar and M. B. Bani-Ismail, "Statistical Classifier of the Holy Quran Verses (Fatiha and Yaseen Chapters)," Journal of Applied Sciences, vol. 5, no. 3, pp. 580–583, 2005.

[40] H. Okuyucu, A. Kurt and E. Arcaklioglu, "Artificial Neural Network Application to the Friction Stir Welding of Aluminum Plates," Materials and Design Journal, vol. 28, no. 1, pp. 78–84, 2007.

[41] M. Al-Abri and N. Hilal, "Artificial Neural Network Simulation of Combined Humic Substance Coagulation and Membrane Filtration," Chemical Engineering Journal, vol. 141, no. 1, pp. 27–34, 2008.

[42] J. Huang, J. Lu and C. X. Ling, "Comparing Naive Bayes, Decision Trees and SVM with AUC and Accuracy," Proc. of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp. 553–556, Melbourne, FL, USA, 2003.

[43] A. Statnikov, C. F. Aliferis, D. P. Hardin and I. Guyon, A Gentle Introduction to Support Vector Machines in Biomedicine: Volume 2: Case Studies and Benchmarks, World Scientific, 2013.

[44] J. Platt et al., "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," Adv. Large Margin Classif., vol. 10, no. 3, pp. 61–74, 1999.

[45] Z.-Q. Zeng, H.-B. Yu, H.-R. Xu, Y.-Q. Xie and J. Gao, "Fast Training Support Vector Machines Using Parallel Sequential Minimal Optimization," Proc. of the 3rd International Conference on Intelligent System and Knowledge Engineering, vol. 1, pp. 997–1001, Xiamen, China, 2008.

[46] L. Wang, Support Vector Machines: Theory and Applications, vol. 177, Springer Science & Business Media, 2005.

[47] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," Proceedings of the 9th ACM International Conference on Multimedia, pp. 107–118, [Online], available: https://doi.org/10.1145/500141.500159, 2001.

[48] M. A. Hearst et al., "Support Vector Machines," IEEE Intelligent Systems, vol. 13, no. 4, pp. 18–28, 1998.

[49] R. Collobert and S. Bengio, "SVMTorch: Support Vector Machines for Large-scale Regression Problems," Journal of Machine Learning Research, vol. 1, no. Feb, pp. 143–160, 2001.

[50] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121–167, 1998.

[51] M. Lalaoui, A. El Afia and R. Chiheb, "A self-tuned Simulated Annealing Algorithm Using Hidden Markov Model," International Journal of Electrical and Computer Engineering, vol. 8, no. 1, p. 291, 2018.

[52] A. El Afia, M. Sarhani and O. Aoun, "Hidden Markov Model Control of Inertia Weight Adaptation for Particle Swarm Optimization," IFAC-PapersOnLine, vol. 50, no. 1, pp. 9997–10002, 2017.

[53] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer and P. A. Torres-Carrasquillo, "Support Vector Machines for Speaker and Language Recognition," Computer Speech and Language, vol. 20, no. 2, pp. 210–229, 2006.

[54] K. M. O. Nahar, M. Al-Shannaq, A. Manasrah, R. Alshorman and I. Alazzam, "A Holy Quran Reader/Reciter Identification System Using Support Vector Machine," International Journal of Machine Learning and Computing, vol. 9, no. 4, pp. 458-464, 2019.

[55] U. Shrawankar, "Techniques for Feature Extraction in Speech Recognition System: A Comparative Study," arXiv Prepr, arXiv1305.1145, [Online], Available: https://arxiv.org/ftp/arxiv/papers/1305/1305. 1145.pdf, 2013.

[56] T. Kinnunen, R. Saeidi, J. Sandberg and M. Hansson-Sandsten, "What Else Is New Than the Hamming Window? Robust MFCCs for Speaker Recognition *via* Multitapering," Proc. of the 11th Annual Conference of the International Speech Communication Association, Japan, [Online], Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.5674&rep=rep1&type=pdf, 2010.

[57] S. Chowdhury, N. Mamun, A. A. S. Khan and F. Ahmed, "Text Dependent and Independent Speaker Recognition Using Neural Responses from the Model of the Auditory System," Proc. of the IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 871–874, Cox's Bazar, Bangladesh, 2017.

[58] R. Bharti and P. Bansal, "Real Time Speaker Recognition System Using MFCC and Vector Quantization Technique," International Journal of Computer Applications, vol. 117, no. 1, pp. 25-31, 2015.

[59] S. A. Majeed, H. Husain, S. A. Samad and T. F. Idbeaa, "Mel Frequency Cepstral Coefficients (MFCC)

Feature Extraction Enhancement in the Application of Speech Recognition: A Comparison Study," Journal of Theoretical and Applied Information Technology, vol. 79, no. 1, pp. 38–56, 2015.

ملخص البحث:

يشير تمييز قراءات القرآن الكريم الى تحديد نوع القراءة ضمن الأنماط المعتمدة (بالعربية: القراءات). لقد بحثت دراسات سابقة عديدة في قواعد قراءة القرآن الكريم المعروفة بأحكام التجويد، التي تطبق من قبل القُرّاء عند القراءة بصوتِ عالٍ. إلا أنه ليست هناك دراسات سابقة تناولت مسألة تتبُّع نوع القراءة. ومن خلال هذا البحث، يمكننا مساعدة الطلبة في تعلُّم قراءة القرآن الكريم بسهولة ودقة عبر التطبيق الدقيق لأحكام التجويد والتمييز بين أنواع القراءة المختلفة.

في هذه الدراسة، تم تطبيق نموذج تمييز مقترح من أجل تمييز نوع قراءة القرآن الكريم من الموجة الصوتية الخاصة بتلك القراءة. وقد بُني النموذج المقترح على ثلاث مراحل؛ الأولى هي مرحلة استخلاص السِّمات من الإشارة الصوتية ووسْمها. أما المرحلة الثانية فهي تدريب النموذج (SVM) على السِّمات المستخلصة والموسومة، في حين تتمثل المرحلة الثالثة والأخيرة في تمييز نوع القراءة بناءً على ذلك النموذج. ولتحقيق ذلك، قمنا ببناء حُزمةٍ تتكون من عشر فئاتٍ كل واحدة منها توسم كنوع من أنواع القراءة. كذلك تم تطبيق مجموعة من خوارزميات تعلم الآلة ومقارنتها. وقد أثبتت النتائج التجريبية تفوُّق نموذج التمييز المقترح القائم على آلة متّجهات الدّعم (SVM) على غيره من الخوارزميات الأخرى التي تم تطبيقها ومقارنتها؛ إذ بلغت نسبة نجاح النموذج المقترح (96%).

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

CLOUD OF THINGS: ARCHITECTURE, RESEARCH CHALLENGES, SECURITY THREATS, MECHANISMS AND OPEN CHALLENGES

Shamsul Haq, Adil Bashir and Sahil Sholla

(Received: 13-Jun.-2020, Revised: 9-Aug.-2020 and 6-Sep.-2020, Accepted: 9-Sep.-2020)

ABSTRACT

In this era of communication and networking technology, Internet of Things adds to the existing technological era and brings revolution to the Information Technology world. Internet of Things consists of interconnected devices which may be digital, physical or mechanical devices equipped with unique identifiers and having the capability to transmit the sensed information to other devices autonomously. Internet of Things is recognized as being composed of resource constraint devices in terms of processing competency, storage capacity and power resources. To cope up these constraints, existing computing technology known as cloud computing can be used to facilitate the Internet of Things system by offloading its processing and storage requirements. In this paper, we have provided the necessity and benefits of Cloud and IoT integration. Further, the paper has identified several research issues that arise due to Cloud-IoT integration. Among the several research issues, it was observed that security and privacy concerns are pivotal in Cloud-IoT integration and need to be addressed to make the integration successful. The core security and privacy threats have been identified and the existing security mechanisms have also been researched in this paper. The paper also highlights open security and privacy research issues in the Cloud-IoT paradigm. This paper can act as a baseline for the research that is needed in the area of security and privacy issues in the Cloud-IoT or Cloud of Things paradigm.

KEYWORDS

Internet of things, Cloud computing, Cloud of things, Security, Privacy, Encryption algorithms.

1. INTRODUCTION

Internet of Things (IoT) is a networking paradigm that connects billions of heterogeneous devices, called Things, within the same backbone, essentially, the Internet. These connected things sometimes referred to as objects, have the ability to generally capture environmental data using dedicated sensors, process the acquired data and communicate the data with other things, within the framework of a smart application [1]. According to the definition of Internet Architecture Board (IAB), IoT is a network of smart things and a way of intelligent communication among an enormous number of connected devices that use a new version of internet protocol (IPV6) providing 2^{128} unique addresses, capable of realizing the actual concept Internet of Things. IoT is swiftly growing and is expected to connect billions of objects in the near future, which requires billions of network addresses. The IoT in whole can be said as an innovation to put together smart things, frameworks and sensors [2]-[4]. The basic building blocks of IoT include hardware, embedded programming and wireless communication technologies. The core of any IoT infrastructure is billions of interconnected devices containing sensors and actuators to sense or detect any physiological or environmental phenomenon. Notwithstanding, to transmit the information that the devices gather, these devices require handling and processing abilities, so that the data can be structured and formatted for transmission. This handling and processing function is commonly performed by a micro-scale integrated circuit; for example, a System-on-a-Chip (SoC) or a Field-Programmable Gate Array (FPGA). Since IoT devices are embedded devices, they are prototyped using competitive micro-scale platforms; for example, Arduino, Phidgets and Raspberry Pi. Prototyping IoT devices using these platforms requires micro-scale controller programming, circuit construction abilities and profound knowledge of hardware communication standards, such as I2C or SPI, that are used to build communication between the micro-scale controller and the associated sensors and actuators. The embedded programs are regularly developed using computer programming languages such as C, C++, Python and JavaScript.

S. Haq, A. Bashir and S. Sholla are with Department of Computer Science and Engineering, IUST, Awantipora, J&K, India. Emails: s.haq266@gmail.com, adilbashir.445@gmail.com and sahilsholla@gmail.com

IoT is becoming an important and globally used technology because of its remote sensing, monitoring and controlling of object (physical and virtual) services across the existing network-infrastructure to enable direct physical world integration with the computer-based systems. It improves the efficiency and accuracy of objects reduces the intervention of humans and provides safety and convenience. To organizations, the benefits offered by IoT include monitoring of the overall business process, improving the experience of customers, saving money and time and making better product decisions. Generally, it is becoming more abundant and important in transportation, manufacturing and utility organizations, as well as in agriculture, home automation, smart cities and smart healthcare [5]-[6]. The layered architecture of IoT consists of five layers; i.e., Perception Layer, Transport Layer, Processing Layer, Application Layer and Business Layer, as shown in Figure 1 [2]-[3].

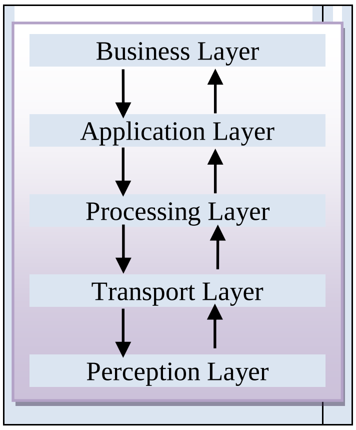


Figure 1. IoT architecture [2]-[3].

1.1 Perception Layer

It is also known as the sensor layer. Its responsibility is to sense the environmental or physiological data of the environment in which it is implemented. The devices are identified and tracked using identification mechanisms, such as RFID tags. Further, the sensor measurements are detected and transformed into electric signals.

1.2 Transport Layer

The services provided by the transport layer include the transmission of information to the processing layer as well as confirming that the information is from the valid user and protecting it from threats using different security protocols. An authentication mechanism based on pre-shared secret keys or passwords is used to verify the valid user.

1.3 Processing Layer

The transport layer sends the information to the processing layer in order to process the collected information. The processing layer removes any extra insignificant information and extracts useful information from the data sent by the transport layer.

1.4 Application Layer

All applications that are using the technology of IoT are defined in the application layer, such as smart cities, smart homes, …etc. The core responsibility of this layer is to provide services to the applications. For each type of application, there may be varying services because of dependent on information collected by sensors.

1.5 Business Layer

It intends the behaviour of an application and is acting as a manager of the whole system. Its responsibility is application control management, business and profit models in IoT. The privacy of users is also managed by this layer [7]-[11]. IoT finds its applications in almost every field, such as industrial automation, supply chain management, intelligent transportation, smart cities and smart healthcare. However, the IoT devices are characterized by constrained resources that hinder their application in sensitive areas where the information needs to be kept secure and safe.

Another important technology, known as cloud computing, enables the development of ubiquitous computing *via* on-demand and convenient access to a configurable shared pool of computing resources, such as network servers, applications, storage and services that can be distributed and released rapidly with less effort and interaction by service providers. Cloud computing is generally divided into two segments; namely, front-end and back-end, based on the architectural viewpoint, as shown in Figure 2 [12].

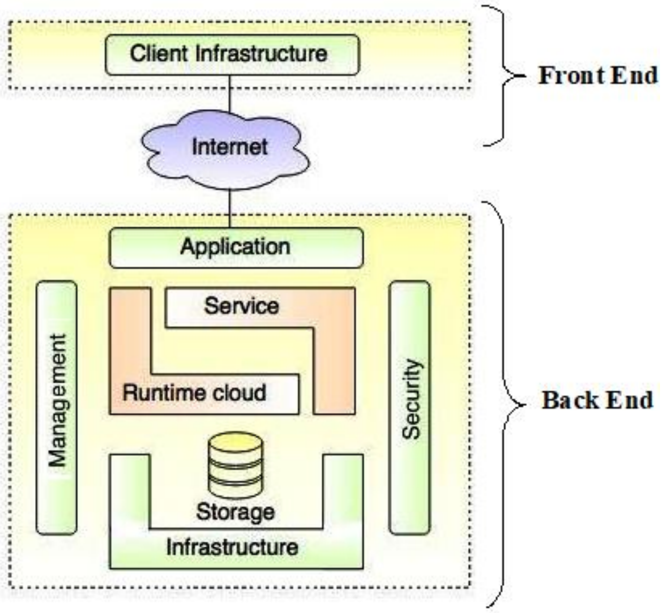


Figure 2. Cloud architecture [12].

The two segments are connected to each other through a network, usually *via* the Internet. The services provided by cloud to its users include Software as a service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [12]-[13]. The benefits provided by the cloud include improvement in performance, massive data handling, minimum issues of maintenance, recovery and backups and scalability [14]-[16].

The greater enhancements in the fields of wireless sensor networks, ubiquitous computing and machine-to-machine (M2M) communication make IoT a more popular and preferred technology. However, IoT devices are resource-constrained, location-specific and inflexible. On the other hand, cloud computing resources are ordinarily area-free (location-independent) and inexpensive, while simultaneously providing fast and precise elasticity. Therefore, to alleviate incompetence in IoT, the Cloud can play a significant role, which necessitates the integration of IoT with the Cloud.

Cloud-IoT is an emerging concept, where the limitations put forth by IoT devices are somewhat addressed using cloud computing services. There are various architectures proposed for Cloud-IoT and most of them focus on data sharing, monitoring, …etc. while using services of the cloud. Though the architectures are varied, some parts among them are common. The simple architecture of Cloud-IoT is shown in Figure 3. The most common elements in Cloud-IoT architectures are:

a) Sensors: Sensors are used to gather data from the deployed environment or objects, such as animals, people, devices, buildings, cities, …etc. The information gathered can be categorized based on sensor type (heterogeneous or homogeneous), sensor methodology (passive or active) or sensing parameters (like body temperature, ECG system, …etc.). The collected data is made available on the cloud [17]-[19].

b) Cloud software: It is responsible for storing and processing information obtained from IoT devices and environments [17]-[19]. It also provides different services for the IoT components, such as monitoring, hosting and managing devices.

c) Network components: These refer to the equipment used for data transmission. Gateways or device drivers are commonly used equipment for the communication between the software components and devices [17]-[18], [21].

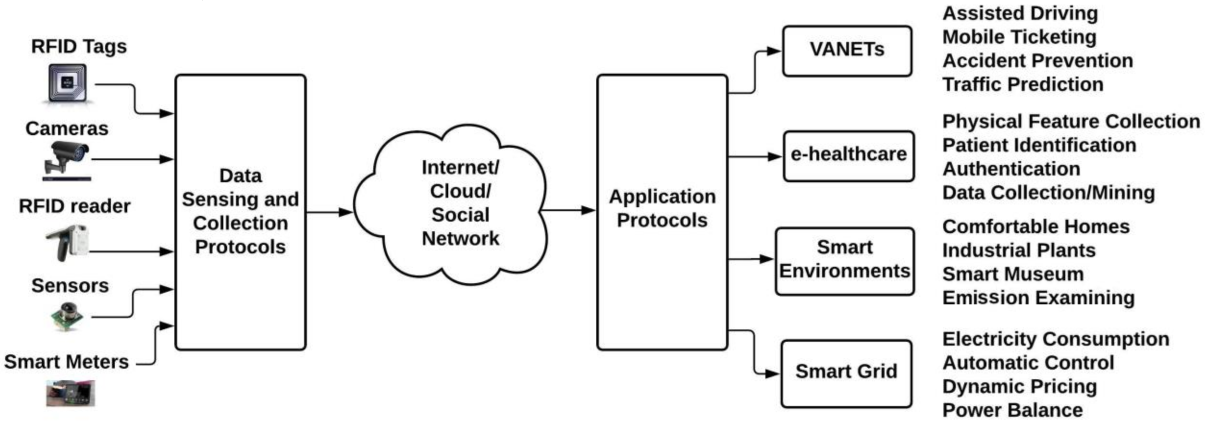


Figure 3. Cloud-based IoT architecture [17].

The rest of the paper is organized in the following sections. Section 2 discusses the importance and benefits of Cloud and IoT integration. Section 3 further explains the research issues in Cloud-IoT integration. Section 4 discusses the security aspects of Cloud-IoT integration. The state-of-the-art security mechanism in Cloud-IoT infrastructure is presented in Section 5 and in Section 6, the open challenges and issues in Cloud-IoT are presented. Section 7 concludes the paper.

2. IoT and Cloud Integration

The current insurgency in the world of information and communication technology (ICT) is equipped by the IoT and services of cloud computing. This offers more open doors for the development of new systems whose main aim is to facilitate the lives of individuals and take out the conventional complexities confronting them. The proliferation of cloud computing and IoT technologies will facilitate new controlling services by handling and processing large volumes of sensory-data streams. Cloud computing is expected to support a wide scope of IoT applications; for example, smart homes, smart cities, smart e-health services, smart buildings, smart grids and so forth [20]. The cloud and IoT are different technologies having different evolutions. The comparisons of cloud and IoT are shown in Table 1. They have complementary characteristics due to which a number of researchers are motivated to test the integration of these two technologies [13]-[14]. The unlimited resource capabilities in terms of processing, communication and storage capacity of the cloud can benefit the IoT and in return, the cloud can get benefited from IoT’s rapidly developing services and getting the opportunity to interact with real-world objects [17]-[18]. Some of the main features and characteristics which relate to both Cloud and IoT include services over the internet, storage over the internet, applications over the internet, energy efficiency and computational capability. The features offered by IoT and Cloud computing are shown in Table 2. Table 2 aims at explaining the enumerated characteristics of cloud technologies that are closely linked to IoT. The observations from Table 2 reveal that IoT characteristics that are more influenced by cloud characteristics are sensors in households and at airports. Concerning cloud computing, the most affected characteristics are services over the internet

Table 1. Comparison of cloud computing with IoT [17].

ITEMS	CLOUD COMPUTING	IoT
Big Data	To manage the enormous big data	Source of big data
Storage capabilities	Unlimited capabilities of storage	Limited or no capabilities of storage
Connectivity	Use of internet for services to deliver	Use of internet for the point of convergence
Processing capabilities	Virtually unlimited capability of computation	Limited capabilities of computation
Characteristics	Ubiquitous (availability of resources from everywhere) The resources are virtual	Pervasive (things are at everywhere) The objects are of real world

and computational capacity. The overall inference that can be drawn is that these two technologies contribute more to each other in many of their characteristics.

Table 2. Contributions of cloud computing and internet of things [18].

IoT Characteristics	**Storage over Internet**	**Service over Internet**	**Applications over Internet**	**Energy Efficiency**	**Computational Capability**
Smart Grids	X	X	-	X	X
Intelligent Transportation	X	X	X	-	X
Sensors installed at homes and airports	X	X	X	X	X
Smart Healthcare	-	X	X	-	X
Engine monitoring sensors	-	X	X	X	X

2.1 Integration Benefits of IoT and Cloud Technologies

IoT is bringing revolution to all the application areas and has made a homogeneous impact on the technology. However, the integration of IoT with cloud put forth several advantages, some of which are presented below [21]-[28].

2.1.1 Scalability

One of the biggest advantages of Cloud-IoT integration is scalability. In case of complex infrastructure of networks, scaling up needs purchasing extra hardware, investigating extra time and undertaking greater configuration and design efforts become difficult. In Cloud-IoT systems, also known as Cloud of Thing (CoT), adding new resources mainly boils down to leasing other virtual servers or extra cloud space which provides the extra benefit of being rapidly implemented. Moreover, the services of the Cloud-IoT platform offer flexibility, providing storage as per requirements and scaling down the number of IoT-enabled systems.

2.1.2 Cost-effectiveness

Large starting upfront investments and expanded implementation risks in the occurrence of in-house IoT systems can be debilitating. Added to that, there is a high concern for the continuous expenses of hardware maintenance (upkeep) and IT personnel (faculty). Fundamentally, scaled-down direct expenses and an adaptable valuing scheme dependent on real utilization urge IoT-based enterprises to change to the cloud. Inside this business model, costs are simpler to anticipate and fewer costs to incur about the hardware equipment failures, which in case of in-house IoT systems may create extra expenses, not to cite business misfortunes resulting from service halts and downtimes.

2.1.3 Improved Processing Capabilities

The limited processing capacity present in IoT nodes and the enormous volumes of data generated by these miniature nodes can be stored, processed and analyzed in the cloud. To find out the solutions, the cloud provides unlimited virtual capabilities of processing and on-demand services or model of usage. There are decision-making and predictive algorithms that can be integrated with the IoT to reduce risks and increase the revenue at a lower cost [26]. Further, the pathways for transfer, storage and maintenance of data are being created by the cloud.

2.1.4 Remote Access (Geographic Bound)

As the growth of the internet is rapid, IoT systems are growing rapidly and are the next step in the near future. This way, various tasks, like monitoring, performance check, data collection and software up-gradations, can be time-consuming and costly processes. However, cloud computing in IoT assists in the immediate accessing and storing of data remotely. This is an essential trait and is not bound geographically and therefore can allocate recourses quickly at different areas. Due to this advantage, there are greater benefits such as that some of the applications can report their status, process the data remotely and send remote messages to inform their administrator about some of the incidents, …etc. [27]-[30].

2.1.5 Data Integration

With the presentation of IoT in business models, organizations are battling with information support. The issue is not just because a lot of information is generated from a wide range of gadgets and devices, but the variety of information that is being generated by smart IoT devices. Furthermore, there is pre-existing traditional information held in these organizations that is being transferred to servers through internet resources. The management of diverse information types is a daunting task for IT designers and executives, because such information is a significant resource of an organization. Cloud computing can help in managing such diverse information from sensors and already held information of organizations by providing a framework with practically no constraints that can scarcely ever be imperilled, because the information is maintained at various separate servers. Along these lines, even in instances of abrupt catastrophes, the cloud will retain the information. Along with these benefits, companies are continuously increasing solutions of clouds as trusted and preferred approaches.

3. RESEARCH ISSUES IN CLOUD-IOT

The transfer of data from the real world to the cloud is made possible by the integration of the cloud with IoT. However, there are various challenges to achieve integration benefits, such as heterogeneity, platforms, services and operating systems, which are particular for the development of new applications. The heterogeneity exacerbates when the approaches of multi-cloud are adopted by the end-users and thus, improving the resilience and performance of applications to services will depend on multi-providers [6]. The big data generated from the expected 50 billion IoT devices in the near future requires having more awareness for its secure communication, access, storage and processing. The important issues that need to be addressed include:

3.1 Interoperability

Interoperability is defined as the ability, due to which heterogeneous devices and platforms can coordinate with each other successfully. It is vital for the interconnection of multiple things together among different networks of communication. If we consider an example of devices in a home automation system that consists of fire detectors, surveillance camera, smoking alarms, entertainment systems, lighting systems, …etc., various protocols are needed for these devices to work in tandem. However, to achieve interoperability, there are various types of challenges to be encountered and dealt with. Some of them are presented below.

3.1.1 Proprietary Ecosystem

It is one of the challenges in interoperability, as proprietary protocols are made by some manufacturers, thus preventing other companies from utilizing them, which makes it difficult to have interoperability.

3.1.2 Cost Constraint

It is another challenge in ensuring interoperable services that are generally faced while designing the gateway solutions. There are certain issues while designing newer protocols, such as the existence of legacy protocols that make it slightly difficult for IoT to use newly designed protocols. Additionally, the technical risks of new protocols may have a higher failure rate.

3.1.3 Scalability

IoT is entering all fields of application, such as transportation, smart buildings, supply chain management, …etc. and the number of devices is increasing rapidly. Therefore, manufacturers are keen to think there would not be an issue of scalability, if for a large number of devices, newer protocols are needed to provide services [31]-[32].

3.2 Connectivity, Compatibility and Longevity

3.2.1 Connectivity

Although the vast number of devices in the IoT network are to be linked in the future, this may contradict with the current structure of the communication protocols and the underlying technologies. Presently, the communication architecture mostly relies on the centralized client-server paradigm to connect the different nodes of the network and so far the authentication and authorization seem sufficient for the current ecosystem of IoT involving hundreds or thousands of devices. But, when there will be billions of devices to join the network, the brokered centralized structure will be required, which in turn becomes the bottleneck for performance and unauthorized activities. These systems would require a tremendous investment in maintaining cloud servers to manage large-scale sharing of information and then all of the systems will go down regularly if the server goes down or is inaccessible. It is projected that future IoT will be based on a decentralized paradigm and cloud servers shall have the responsibilities of gathering and analyzing the enormous sensed data. Other solutions may involve the use of a peer-to-peer communication model, where devices can identify and authenticate each other directly and can exchange information without involving brokers. This decentralized model will have its challenges, especially in terms of security, but that can be met with emerging technologies, such as blockchain technology.

3.2.2 Compatibility

IoT involves different technologies with varied compatibilities, which makes it difficult for any procedure to compete for becoming standard. It requires extra hardware and software to connect the devices. The other compatibility issues are from the non-unified services of the cloud, diversities of operating systems and firmware and lack of standard M2M protocols among the IoT devices.

3.2.3 Longevity

The persistence of the technologies used to create Cloud-IoT systems is essential for the smooth functioning of the deployed systems. In the next few years, some of these technologies will eventually become redundant, potentially rendering the nodes that implement them useless or ineffectual. This is mainly important, as compared to generic devices of computing having a life span of few years, appliances of IoT (like TVs, smart fridges,…etc.) tend to remain in service for much more longer and should be functional even if the manufacturer of theses gadgets goes out of services [32]-[34].

3.3 Standards

The technological standards that are used for the proper functioning of devices and to deliver services effectively include data collection standards, networking protocols, communication standards and the procedures used for handling, processing and storing of data obtained from servers. This type of aggregation is to increase the data value by increasing the scope, scale and frequency of data available for analysis. The challenges in standardization include developing standards for unstructured data handling, leveraging new tools of aggregation by technical skills. The structured type of data, for example, is stored in the relational database and queried through MySQL. However, the unstructured data is stored in different types of databases consisting of NoSQL without any standard approach of the query. In terms of technical skills, companies often face the challenges of shortage of talent to make strategy, plan, execute and maintain the systems to leverage unstructured big data [17].

3.4 Security

The fast progressions made in IoT technology are changing lives by connecting a vast number of user gadgets to the internet and thereby controlling them remotely. Along with different applications, such as e-health applications based on Cloud-IoT, frameworks are more efficient and offer better services to the users. However, the usage of Cloud-IoT-based systems demands high security of the data that lies within Cloud-IoT infrastructure, as it involves user’s private data. Security and privacy have never been as vulnerable as these are currently, with a vast number of systems sending and receiving immense amounts of information wirelessly. Therefore, researchers must focus on developing and enhancing existing privacy and security solutions for Cloud-IoT-based frameworks; for example, schemes of automatic identification, watermarking, active smart-monitoring and verification of fingerprint schemes [35]. Among the various security issues, one of the important and concerning challenges is to minimize IoT node resources that are consumed by security protocols and to reduce the security vulnerabilities, attacks and threats. Another essential security issue is to provide the rules for authorization and the polices to ensure that sensitive data is accessed only by authorized users, which is pivotal to maintain the privacy of users, particularly when there is a need for integrity to be guaranteed. Data integrity is concerned as the vital element that affects not only the quality of services, but also the majority of security and privacy issues related to it, such as outsourced data, legal aspects, large scale, monitoring and performance. Additionally, several other issues exist, such as lack of trust, location of physical data and information concerning Service Level Agreements (SLAs) when IoT applications are moved to the cloud. The leakage of information can also happen due to multi-tenancy, which can result in unauthorized effects. Furthermore, key-based cryptographic algorithms, such as public-key algorithms, cannot be applied to IoT devices because of the imposition of constraints in their processing power, battery capacity and memory. Specific attention to the growing challenges is also required; for example, there is a possibility of new attacks, such as SQL injection, cross-site scripting, session hijacking and side-channel attacks, to occur. Moreover, there is much vulnerability that includes virtual machine escape and session hijacking, which are problematic to Cloud-IoT infrastructure [35]-[36]. These issues need to be addressed before the Cloud-IoT paradigm will be fully implemented and adopted by the general user group.

4. SECURITY AS A RESEARCH ISSUE IN CLOUD-IOT

Among the research issues discussed above, security is pivotal, because the connected devices having internet connectivity monitor the user devices continuously, which may harm user privacy if that data is leaked to unauthorized users. A large amount of personal data is generated by smart IoT devices; therefore, users need to know that their information is secure and safe and the business has legal responsibilities to keep information secure. IoT systems facilitate the association of both large and small frameworks together and utilize the internet for communication. Users want to be sure about the security of their IoT gadgets before adopting them fully in their usage. IoT frameworks are inherently vulnerable to outside attacks, because the IoT systems use conventional networks to connect everything wirelessly. Researchers are actively working to find viable solutions for many security and privacy issues in Cloud-IoT systems; however, these issues need further investigation, so that user privacy is not compromised [35]. The various security issues that arise in Cloud-IoT include the common and important issues of Confidentiality, Integrity, Availability of sensed data and the Authentication of devices and data itself. Cloud-IoT is a growing technology and is not much more developed to overcome many issues yet, including security. There is much importance of security when developing the solutions of Cloud-IoT, as there are possibilities of many attacks to happen in the development phase. Without addressing security issues properly, users hesitate to adopt the Cloud-IoT technology in their day-to-day life, as it can harm their privacy; for instance, in a smart health care system that needs to have time-to-time valid data of patients for their continuous observation and if this data is damaged by attackers, this can cause serious outcomes, such as wrong medication leading to the death of a patient. Similarly, in smart homes, personal data may be breached for any harm, intelligent transportation may be hijacked to cause accidents, …etc. Security in Cloud-IoT is a core issue that needs to be addressed, because the private data available at the cloud or in-transit to the cloud from IoT devices can be exploited by hackers, leading to unauthorized effects.

4.1 Threats to Security and Privacy in Cloud-IoT

The integration of IoT and cloud brings a lot of vulnerabilities due to the involvement of user-specific miniature devices and associated limitations. Privacy preservation is always a fundamental human right and in a business context. It is said to be a protection of customer information to use it more appropriately. The security and privacy of the information used in case of business entities need to follow the application laws, policies, standards and the process by which personal information is being managed. In this notion of security, it is referred to as information security defined by ISO 27001 standards for the preservation of confidentiality, integrity and availability (CIA) of information. Non-repudiation, reliability, accountability can also be deliberated as need-based security [38]-[39]. Some of the most common threats to security and privacy are illustrated in Figure 4 [40]-[41]. The brief descriptions of these threats are presented below.

4.1.1 Threats to Security

Communication Threats: The communication channel may be abused by attackers and intruders to launch various attacks. The following threats are likely to occur in this category:

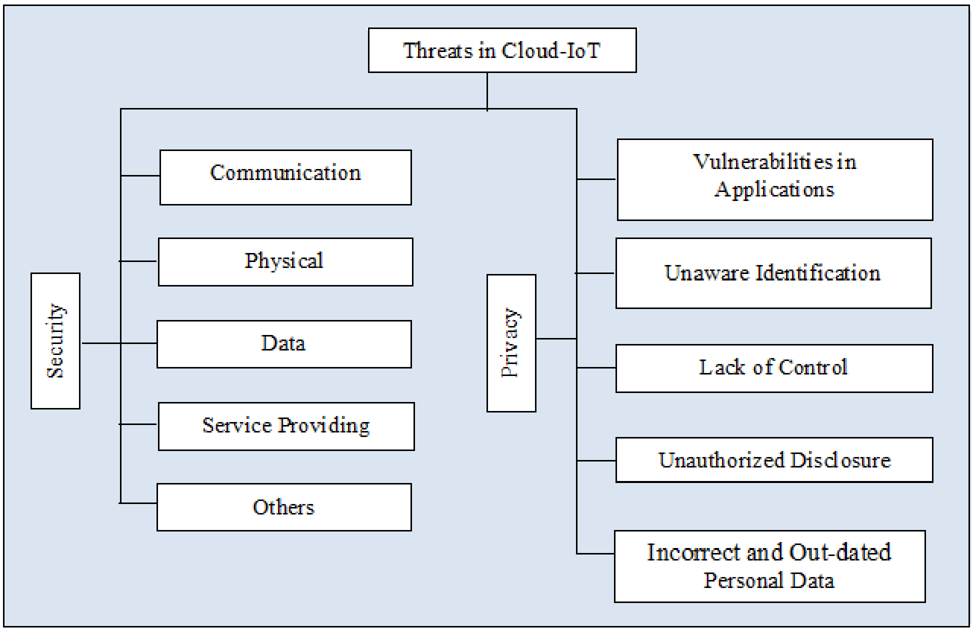


Figure 4. Security and privacy threats in cloud-IoT [40].

a) Denial of Service: Denial of Service (DoS) can be launched over the Cloud-IoT infrastructure in order to reduce the execution of the expected functional capacity of the network, through resource exhaustion, hardware failure and software bugs. DoS attacks are more prominent to the IoT devices in Cloud-IoT infrastructure because of the inherent resource limitations in these devices.

b) Eavesdropping: It refers to the interception of private communications by unauthorized users in real time. Attackers can gain access to the communication channel in order to overhear the secret communication among different network entities.

c) Spoofing Attack: It is an attack where an attacker impersonates and pretends to be an attacker to gain access to restricted and privileged services and bypass existing security and authentication mechanisms. This attack is usually a starting point for a more impactful attack, in most cases, a DoS attack. Such attacks are common to internet-connected devices and therefore make an important attack scenario in the Cloud-IoT system as well which needs to be addressed [41]-[44].

d) Man-in-the-Middle Attack: It is the common cybersecurity attack that establishes security credentials with the sender by impersonating itself to be the receiver. The sender expects to be communicating with the receiver device; however, the information exchange is happening with the attacker that is present between sender and receiver. The attacker then sends the altered messages to the receiver.

e) Replay Attack: This attack relies on an insecure network, in which the attacker captures the data packets and then forwards them at a later time to produce unauthorized effects at the receiver device. The transmission of data is interrupted or replicated by the malicious party, who intercepts the data and retransmits it. A replay attack is possible on a communication protocol when data freshness is not provided. Thus, the network could be secure with respect to authentication, confidentiality and integrity, but does not provide data freshness to mitigate replay attacks.

Physical Threats: These refer to the incidence of harming the devices physically to damage the network devices, resulting in the loss of system and information. The following attacks can be launched under this category:

a) Device Capture: The attackers can capture legitimate IoT devices in order to extract the information held in these devices before it has been transmitted to the secure system for storage. The attackers can also extract the security keys and the shared secrets at IoT devices. The legitimate IoT devices are destroyed by damaging their radio-module and by deleting their memory, which can result in severe damage to Cloud-IoT infrastructure.

b) Node Damaging: Having easy access to physical devices, an attacker can damage any of them physically, which makes them unable to sense and transmit the data. DoS attacks can also be launched if more devices are damaged by an attacker, so that the entire Cloud-IoT system will become useless and incapable of providing any type of services.

Data Threats: Data threats are considered as common threats for every internet user. The most common threats are: spams, disabling security settings, data corruption and data stealing. The following are the likely threats under this category:

a) Threats during Retrieval, Transfer and Storage of Data: If an attacker gets physical access to an IoT device, then he/she can access the raw sensed data available at devices using micro-probing or reverse engineering techniques. Usually, Cloud-IoT needs to transfer the data at IoT devices to the cloud for storage and processing; therefore, there is a greater risk of data tampering during the transfer of data to the cloud [45].

b) Unauthorized Device Deployment: The attackers can deploy their own devices in the network and send false or infected data to the cloud, resulting in corrupting the entire data that is stored at the cloud. Therefore, the establishment of device authenticity in Cloud-IoT systems is mandatory and if any device fails to prove its authenticity, the data from that device shall not be accepted.

c) Data Loss and Leakage: Events happening accidentally, such as fire, deletion of data by the service provider, earthquakes, …etc. causing loss of critical data. The data can be leaked to unauthorized users accidentally, which can be protected by encryption mechanisms.

d) Data Breach: In this attack, the data can be accessed by unauthorized entities from inside or outside of the system. All types of Cloud-IoT data do not have the same level of sensitivity, as some financial data is more sensitive than other publicly available data and therefore needs to be more protected.

Provision of Service Threats: In Cloud-IoT, many services are used to ensure the operations to happen smoothly, but the threats related to them include the following:

a) Unidentified Users: Services provided by the cloud must ensure that unidentified and unauthorized users cannot gain access to the data being sensitive or not; otherwise this may result in corrupting or authorization of the entire Cloud-IoT infrastructure [42].

b) Identity Theft: In this attack, attackers can access the services and resources, which were otherwise restricted to the user, by gaining access to the credentials of valid attackers that can make the victim accountable for the attacker's actions [46].

c) Compromising Interfaces: It is considered in the top threats of Cloud-IoT, because the APIs are always distributed by the cloud providers to help consumers retrieve data and get access to other services. If the interfaces are not well protected, the attacker can easily get their weakness to be exploited and so -by attacker's data- launch fraudulent services [45]-[47].

Other Threats: Various other threats are not related to the defined categories and some of them are presented below:

a) Malicious Insider: In Cloud-IoT systems, sometimes the attacker having valid authentication and authorization credentials may harm or attack the secret information at network devices by perpetuating the malicious activity on the network. This way, he/she can exploit the access to abuse services.

b) Shared Technology: In Cloud-IoT systems, there exist several shared resources that can be used remotely. Using the shared resources through virtualization can allow access of other Virtual Machines (VMs) of other users, which occurs due to vulnerabilities in VM monitor that may be exploited by malicious users to gain access to the other valid users' VMs.

c) Cloud Computing Abuse: The biggest advantage of cloud computing is that a user can have huge computing power available that is allowed by the organization, which can assist malicious users to get an opportunity to launch varied attacks. A single attacker can even get many resources of computing on-demand to launch a DoS attack to other cloud service providers [48].

4.1.2 Threats to Privacy

The evolution of Cloud-IoT emerges new ways of interaction that concern various privacy and security issues based on technologies and features that are used to deliver Cloud-IoT services. Many threats can be exploited to harm the privacy of users in Cloud-IoT systems. Some of the likely threats are presented below:

a) Vulnerability in Applications: If companies do not consider the vulnerabilities in delivering the application patches or even complete applications, the hackers or attackers can exploit these vulnerabilities to enter the system and create unauthorized effects.

b) Unaware Identification: IoT devices can be used to collect user data without their knowledge, which can be achieved by using undisclosed small-sized cameras or sensors in users' devices or surrounding areas. The data collected can be used to identify the user and his/her associated attributes, which is the real threat to user privacy [39].

c) Lack of Control: Once the data is collected and uploaded on the cloud, sometimes it is possible to have either limited access or no access to it and in other words, it can be said that the control over data is sometimes lost. The ubiquitous process of sense makes it difficult for users to give their consent to collect data or the actions to be performed after the collected data has been processed and analyzed. Additionally, it becomes a challenging task to create rules for access control to protect privacy. To have keen attention to the preservation of privacy in any system, consent is considered as a major requirement for the collection, storage and processing of personal information [49].

d) Unauthorized Disclosure: The use of cloud infrastructure might impact the users when cloud providers experience difficulties to get the consent about user data collection and processing which may result in unauthorized disclosure of sensitive data.

e) Incorrect or Out-dated Personal Data: It is to be resolved if there is any out-dated or incorrect data in the system; for instance, a patient in an e-health system has been diagnosed with some illness which gets cured after some time. If this kind of information is not updated in the database, the treatment in accordance with the previous report will be harmful to the patient. Similarly, companies should maintain data accurately and update it frequently.

5. STATE-OF-THE-ART SECURITY MECHANISMS IN CLOUD-IOT

The security of Cloud-IoT systems depends on the type of application they are used in. For example, in a smart home application, the security of the latter relies on various things, such as the security of the devices themselves, the security of the wireless infrastructure where these devices are connected (e.g., the home Wi-Fi network), the security of the wired network that connects the smart home to the Internet and the security of the cloud service that the homeowner is subscribed to. Similarly, in e-health application, the security of such a system depends on the security of network infrastructure where medical sensors are deployed, the mobility of patients, the cloud service to which patients and doctors are subscribed to, the security of the medical sensors and the devices that doctors use to monitor and prescribe medicines to patients and the security of network infrastructure through which electronic health records are exchanged. In all applications of Cloud-IoT systems, the user data needs to be protected from attackers and thus, the security solutions are developed so that the sensitive data is protected from attackers. The IoT security solutions involve the secure architecture of multiple levels that use important features of security in IoT across four different layers which are briefly defined below:

a) Device Level Security: Device level security refers to the hardware level solutions of IoT. The security components in this level include chip security, secure booting, device identity and authentication and physical security.

b) Communication Level Security: Communication level security refers to the security of the connection medium through which data is transmitted and received. The security components in this layer include access control, end-to-end encryption, intrusion detection and preventions and firewalls.

c) Cloud Level security: It refers to securing the software backend solution of IoT. Cloud-IoT providers are expected to provide security from major breaches of data itself. The security components in this layer include platform security, data at rest and verification of application integrity.

d) Lifecycle Management Securities: Lifecycle management securities refer to securing the continuous processes that are required to keep the IoT solution's security up-to-date. The security components in this layer include policies and auditing, risk assessment and secure decommissioning.

The security mechanisms based on cryptographic protocols need to play an important role at all levels of security enhancement. The detailed view of security encryption mechanisms and the protocols being used are discussed below.

5.1 Encryption Mechanisms

Most of the components of Cloud-IoT systems, such as IoT devices, storing devices and cloud are vulnerable to different attacks. Attackers can find the location of any network node or device on the network and can easily harm it. These situations can be avoided using encryption mechanisms, such as enciphering the data and its storage location and transferring the encrypted data among devices instead of unencrypted data [50]. Stored data in data warehouses can also be attacked; therefore, the need for strong encryption mechanisms is required. Various secure and popular cryptographic algorithms are implemented for internet security and some of them are shown in Table 3 [51]-[52].

Table 3. Cryptographic algorithms suite [51].

Algorithms	Purpose
Rivest Shamir Adelman (RSA), Elliptic Curve Cryptography (ECC), ECDSA	Confidentiality, Digital Signature
Advanced Encryption Standard (AES)	Confidentiality
HMAC, SHA-3, BLAKE-3	Integrity
Diffie-Hellman (DH)	Key Agreement

AES is given the first option of all the standards, as it can be used at all layers of IoT for imparting security, while ECC is viewed as another primitive used at the physical layer, network layer and application layer. The protocols that employ AES as a security construct include Constrained Application Protocol (CoAP), which is used as an application layer protocol for the Internet of Things, Bluetooth-Low-Energy version 4.2 (BLE 4.2), Internet Protocol version 6 (IPv6), 6LoWSec and 4G [53]-[54]. The different security policies are chosen in accordance with the application demands, such as whether end-to-end encryption is required or not. End-to-end encryption provides high-level security, wherein the sender and the receiver can only read the message content and none in the middle can get the message content. In traditional networks, TLS/SSL and IPSec protocols are commonly used to provide authentication, integrity and confidentiality services to communication messages. IPSec is designed to provide security at the network layer either in transport or tunneling mode, whereas the TLS/SSL protocol is used to provide security services at the transport layer. TLS/SSL or IPSec protocols can be used by the applications to access the internet *via* encrypted details of authorized users. Weak APIs and interfaces are attractive attributes for attackers to capture or sniff packets. The most important security construct for users in Cloud-IoT is the availability of network services. A potential attack to availability is the DoS attack, which is launched through the flooding of packets to exhaust network resources. Researchers believe that Cloud-IoT is more vulnerable to DoS or Distributed Denial of Service (DDoS) attacks, as it is shared by many users, which can be reduced by monitoring user requests. Before processing the request, prior identification of undefined requests or duplicated messages shall be erased [51], [55]-[58].

5.2 Different Ways of Handling Cloud-IoT Security

There are different methods to handle the security concerns which mainly rely on cryptographic protocols and some of the ways to handle Cloud-IoT security challenges include:

a) Cloud-IoT security analytics: It involves collecting, correlating and analyzing data from various sources that can help security providers identify threats and nip them up.

b) Public Key Infrastructure use: It includes the set of policies, hardware and software means that are needed for the creation and distribution of digital certificates, which are essential components for various public-key schemes. To be an effective solution for Cloud-IoT security, this method has proven successful over the years. Some of the Public Key Infrastructure (PKI) methods are used for the management of private or public keys and X.509 digital certificates.

c) Ensuring protection of communication: The communication of sensitive information in IoT needs to be protected from hackers and attackers, which can otherwise lead to unauthorized effects. Cryptographic algorithms, such as AES, ECC and RSA are the most widely used encryption algorithms.

d) Ensuring authentication of devices: Device authentication is essential for ensuring that malicious data is not injected into the network by malicious nodes, which can result in damaging the crucial information in the network. Two-factor authentication, digital certificates and biometrics are the basis of authentication to reduce vulnerabilities.

5.3 Existing Security and Privacy Solutions of Cloud-IoT

The existing security techniques proposed in the literature to ensure security and privacy in Cloud-IoT systems are presented in this sub-section.

Authors in [59] presented that developing confidential infrastructure for Cloud-IoT applications is very expensive in comparison to the low-cost infrastructure of the public cloud, due to the large amount of data generated by IoT. Therefore, more public clouds are used for processing tasks even in case of sensitive data, which leads to increased concern for maintaining the confidentiality of data. One of the ways to mitigate the concern of confidentiality is to encrypt the data at the source and then send it to the cloud for storage purposes only. One of the promising approaches proposed to overcome this bottleneck is Partial Homomorphic Encryption (PHE). In this paper, a scheme for confidentiality preserving continuous query execution in an un-trusted cloud through API initiative that allows programmers to focus on the analysis of automatic homomorphism, the logic of applications and original techniques of compilation, has been proposed. In Zhu et al. [60], the scheme of data integrity is proposed with the combination of ZSS signature and is related to security, privacy and scalability to meet the requirements of computation and storage functions of analytical applications with big data. The remote integrity is implemented while using the ZSS signature. With the use of the ZSS signature, the computational overhead is reduced compared to BLS algorithms. This represents the solution with less overhead in terms of communication and computation than in current RSA and BLS-based data integrity solutions. Authors in [61] proposed a security framework for the Cloud of Things (CoT) that addresses some of the identified security issues in the existing CoT environments. The proposed framework provides several advantages in terms of efficient resource usage, data prioritization, data delivery timeliness and an adequate security level to sensitive data. A confidentiality-preserving system for CoT has been proposed in [62], which uses the Partial Homomorphic Encryption (PHE) mechanism to encrypt the data. The proposed system enables programmers to concentrate on application logic, compilation mechanisms, homomorphism evaluation and optimized resource usage. Authors in [63] have designed a deep reinforcement learning-based malware propagation model. The developed model has been assessed for energy consumption *vs.* number of nodes, average infections over time, node mobility over time period and propagation speed.

In [64], the proposed architecture to achieve availability is ascertained through the execution system based on the Open-STACK. To ensure availability, a template-based cloud framework has been proposed, which can configure fault identification and recovery measures automatically according to different services and features. According to the characteristics and services, proposed method applications were allowed by the templates and the feasibility methods were demonstrated with the existing architecture *via* comparison. In [65], an authentication scheme has been proposed, in which the biometric parameters are combined with the user credentials. The additional key is generated for the ECC algorithm for improving its security level. Normally, in ECC, only two keys are created that are public and private; however, in improved ECC, an additional secret key is generated. This additional secret key achieves the requirements of security, like low encryption, computation and decryption time overhead. Authors in [58] proposed the concept of secure trusted things aiming to reduce the security and privacy concerns in Cloud-IoT systems. It includes an encryption mechanism that involves less overhead. Authors in [66] have proposed a lightweight security scheme for IoT, wherein the energy-efficient and simple cryptographic operations are used. Authors in [67] proposed a security scheme for smart home systems based on Cloud- IoT infrastructure. It proposes group key management for smart home system. Here, the proposed scheme ensures secure data transfer *via* symmetric key cryptography. The analysis of the proposed security scheme depicts that it is easy to implement, energy-efficient and flexible.

In [68], integration of secure and intelligent security architecture is proposed for the Cloud of Things, in which users are able to access applications in the cloud. Elliptic Curve Cryptographic (ECC) has been used to provide security services. Authors in [65] designed biometric authentication for a multi-cloud sever environment. The core building blocks in their scheme are biometric-based hashing and ECC. In [69], the security and privacy challenges are investigated and discussed by introducing the fog computing in IoT. In this investigation, the authentication issue has been considered as the main challenge with the context of Cloud and Fog computing that is coupled with the applications of IoT. In [68], an adaptable model has been proposed for securing communications in Cloud-IoT systems in contrast to existing pre-configured solutions. It defines the operations of secure communication to agree dynamic and autonomous security protocols and keys of cryptography. Authors in [70] have analyzed the effect of mobility on the authentication and have used Forwarding First (FF) Protocol and Authentication First (AF) Protocol in their analysis. The results depict that mobility affects these protocols in terms of delay and energy consumption. In [57], privacy preserving in message forwarding scheme is constructed for Cloud-IoT systems that are intended to improve efficiency and privacy of transmission. They have developed the architecture of the cloud server having two layers in order to improve the efficiency of communication of clients. In [18], the secure Cloud-IoT method is proposed. The authors conducted the survey based upon the security issues in technologies involved in both Cloud and IoT. After discovering the benefits of cloud and IoT, the authors have surveyed the security challenges in the Cloud-IoT system and proposed a method that improves privacy and security issues in Cloud-IoT. In [72], a list of security solutions, such as the use of private cloud with the parameters of enterprise, session container use, encrypted content and cloud access broker visualization of security at the run-time, have been presented. These solutions are used for different application demands and are expected to improve the overall security of Cloud-IoT-based systems.

6. PRIVACY AND SECURITY OPEN CHALLENGES

It is imperative to manage access, communication and use of available resources of IoT and make efficient protocols and standards for such resource-limited devices. The information that IoT devices gather shall be protected from unauthorized access. There is a need for crucial technologies to protect the individual's privacy and security in the context of Cloud-IoT while having reliable and efficient communication between IoT and cloud infrastructure. It is observed that the benefits of the integration of cloud and IoT have also generated new sets of research challenges. Therefore, there is a further requirement of transformation in technologies of cloud to manage the flow of data and ownership of data source within Cloud-IoT. The data is also accessed by third parties while having virtualization of IoT resources, a new type of interrogations with the regard to data ownership and trustworthiness of data is needed to be addressed. Furthermore, eavesdrop and monitoring of people without their knowledge and consent is another serious problem.

In Cloud-IoT systems, *ad-hoc* connections and backend cloud communication are commonly occurring activities which demand security. Authors in [73] presented that there is a need for an agreement protocol for secure communication that allows the communicating entities to have a mutual agreement based on keys and cryptographic algorithms. In the Cloud-IoT environment, there is a need for adaptable and flexible agreement mechanisms because of storage, bandwidth and computational limitations in IoT. The major obstacles and challenges to practical security in Cloud-IoT systems are given below:

a) Dynamic Cycle of Activity: Different roles and functions can be taken up by connected IoT devices depending on the security challenges in Cloud-IoT systems. The data may be directly transmitted to other devices or cloud servers.

b) Interaction of Heterogeneity: Cloud infrastructure is provided by different manufacturers who use different sets of protocols, technical requirements and standards, which puts hindrances for the interoperability among connected device platforms as well. Because of heterogeneity of the protocols and technological features of the interconnected devices, it is important to implement new security algorithms in order to ensure safe communication of sensitive data.

c) Provision of Antivirus: Antiviruses are usually used in traditional networks to protect personal computers from attacks and malware. These are memory-consuming and put a great challenge for being used in resource-limited IoT devices.

To make Cloud-IoT a successful technology, various identified research issues need to be addressed by the research community to make Cloud-IoT globally adopted.

7. CONCLUSION

The new and growing technology known as Cloud-IoT or Cloud of Things (CoT) is going to make a huge impact in the future. Both these technologies vary in their characteristics and features but aggregating them together brings several benefits, such as minimization of effort, less costs to incur on hardware, interacting with real-world entities and the like. IoT generates massive data that cannot be stored in IoT device memory or on simple servers; therefore, bringing the cloud into the picture solves big data issues in IoT. On the other hand, IoT assists the cloud to be able to interact with real-world objects. However, the integration of the two technologies brings several research issues that have been highlighted in this paper and the pivotal research issue that has been observed is the security issue that arises due to the amalgamation of cloud and IoT technologies. In this paper, different threats to security and privacy have been identified and the relevant existing security mechanisms have been presented. Further, open security and privacy issues have been identified, which requires further research efforts in order to address these issues. This paper can act as a baseline for research needed in the area of security and privacy issues in the Cloud of Things paradigm.

ACKNOWLEDGMENTS

This research work is funded under the seed grant initiative of the TEQIP-III project currently being implemented at the Islamic University of Science and Technology, Awantipora, Jammu and Kashmir, India.

REFERENCES

[1] K. Lounis and M. Zulkernine, "Attacks and Defenses in Short-range Wireless Technologies for IoT," IEEE Access, vol. 8, pp. 88892-88932, DOI: 10.1109/ACCESS.2020.2993553, 2020.

[2] S. Kumar, P. Tiwari and M. Zymbler, "Internet of Things Is a Revolutionary Approach for Future Technology Enhancement: A Review," Journal of Big Data, vol. 6, Article no. 111, 2019.

[3] O. Mashal, T-Y. Alsaryrah, C-Z. Chung, Z. Yang, W.-H. Kuo et al., "Choices for Interaction with Things on Internet and Underlying Issues," Ad Hoc Networks, vol. 28, pp. 68–90, 2015.

[4] O. Said and M. Masud, "Towards Internet of Things: Survey and Future Vision," International Journal of Computer Networks, vol. 5, no. 1, pp. 1–17, 2013.

[5] F. Firouzi, K. Chakrabarty and S. Nassif, "Intelligent Internet of Things: From Device to Fog and Cloud," Springer, Cham, DOI: https://doi.org/10.1007/978-3-030-30367-9, 2020.

[6] H. F. Atlam, A. Alenezi, R. J. Walters and G. B. Wills, "An Overview of Risk Estimation Techniques in Risk-based Access Control for the Internet of Things," Proc. of the 2nd International Conference on Internet of Things, Big Data and Security, pp. 1–8, [Online], available: https://www.scitepress.org/papers/2017/62926/62926.pdf, 2017.

[7] S. Rabhakar, "Network Security in Digitalization: Attacks and Defence," International Journal of Research in Computer Applications and Robotics, vol. 5, no. 5, pp. 46–52, 2017.

[8] B. Ali and A. I. Awad, "Cyber and Physical Security Vulnerability Assessment for IoT-based Smart Homes," Sensors, vol. 18, no. 3, Article no. 817, 2018.

[9] A.Sanzgiri and D. Dasgupta, "Classification of Insider Threat Detection Techniques," Proc. of the 11th Annual Cyber and Information Security Research Conference, p. 25, ACM, Oak Ridge, USA, 2016.

[10] P. Sethi and S. Sarangi, "Internet of Things: Architectures, Protocols and Applications," Journal of Electrical and Computer Engineering, vol. 2017, Article ID 9324035, pp. 1-25, DOI: 10.1155/2017/9324035, 2017.

[11] T. Qiu, N. Chen, K. Li, M. Atiquzzaman and W. Zhao, "How Can Heterogeneous Internet of Things Build Our Future: A Survey," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2011-2027, DOI: 10.1109/COMST.2018.2803740, 2018.

[12] G. Daly et al. "Cloud Customer Architecture for IoT," Cloud Standards and Customer, Council Whitepaper, [Online], Available: https://www.omg.org/cloud/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf, 2016.

[13] Z. Qureshi, N. Agrawal and D. Chouhan, "Cloud Based IoT : Architecture, Application, Challenges and Future," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 3, no. 7, pp. 359–368, 2018.

[14] J. Zhou et al., "Cloud Things: A Common Architecture for Integrating the Internet of Things with Cloud Computing," Proceedings of the IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Whistler, BC, pp. 651-657, DOI: 10.1109/CSCWD.2013.6581037, 2013.

[15] I. Odun-Ayo, C. Okereke and E. Orovwode, "Cloud Computing and Internet of Things: Issues and Developments," Proceedings of the World Congress on Engineering (WCE 2018), vol. I, London, U.K., [Online], Available: http://www.iaeng.org/publication/WCE2018/WCE2018_pp182-187.pdf, July 2018.

[16] D. Salvatore, M. Giovanni and P. Antonio, "A Utility Paradigm for IoT: The Sensing Cloud," Pervasive and Mobile Computing, vol. 20, pp. 127-144, [Online], Available: https://doi.org/10.1016/j.pmcj.2014. 09.006, 2015.

[17] H. Atlam, A. Alenezi, A. Alshdadi, R. Walters and G. Wills, "Integration of Cloud Computing with Internet of Things: Challenges and Open Issues," Proc. of IEEE International Conference on Internet of Things (iThings) and IEEE Green Comp. and Com. (GreenCom), pp. 670-675, Exeter, UK, 2017.

[18] S. Christos, E. P. Kostas, K. Byung-Gyu and G. Brij, "Secure Integration of IoT and Cloud Computing," Future Generation Computer Systems, vol. 78, no. 3, pp. 964-975, 2018.

[19] F. Alhaidari, A. Rahman and R. Zagrouba, "Cloud of Things: Architecture, Applications and Challenges," Journal of Ambient Intelligence and Humanized Computing, [Online], Available: https://doi.org/10.1007/s12652-020-02448-3, 2020.

[20] M. B. Yassein, I. Hmeidi, A. Alsmadi and M. Shatnawi, "Cloud Computing Role in Internet of Things: Business Community Survey," Proc. of the 11th Int. Conf. on Information and Communication Systems (ICICS), pp. 343-348, DOI: 10.1109/ICICS49469.2020.239533, Irbid, Jordan, 2020.

[21] D. Kelaidonis, A. Rouskas, V. Stavroulaki, P. Demestichas and P. Vlacheas, "A Federated Edge Cloud-IoT Architecture," Proc. of IEEE European Conference on Networks and Communications (EuCNC), pp. 230-234, DOI: 10.1109/EuCNC.2016.7561038, Athens, Greece, 2016.

[22] L.Celic and R. Magjarevic, "Seamless Connectivity Architecture and Methods for IoT and Wearable Devices," Automatika. vol. 61, pp. 21-34, 2020.

[23] T. Bhattasali, R. Chaki and N. Chaki, "Secure and Trusted Cloud of Things," Proc. of the Annual IEEE India Conference (INDICON), pp. 1-6, Mumbai, India, 2013.

[24] S. Kamburugamuve, L. Christiansen and G. Fox, "A Framework for Real Time Processing of Sensor Data in the Cloud," Journal of Sensors, vol. 2015, Article ID 468047, pp. 1-11, 2015.

[25] L. Hou et al., "Internet of Things Cloud: Architecture and Implementation," IEEE Communications Magazine, vol. 54, no. 12, pp. 32-39, DOI: 10.1109/MCOM.2016.1600398CM, Dec. 2016.

[26] R. K. Dwivedi, S. Singh and R. Kumar, "Integration of Wireless Sensor Networks with Cloud: A Review," Proc. of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 114-119, DOI: 10.1109/CONFLUENCE.2019.8776968, Noida, India, 2019.

[27] S. M. Babu, A. J. Lakshmi and B. T. Rao, "A Study on Cloud Based Internet of Things: CloudIoT," Proc. of IEEE Global Conference on Communication Technologies (GCCT), pp. 60-65, DOI: 10.1109/GCCT.2015.7342624, Thuckalay, India, 2015.

[28] M. Korunoski and M. Gushev, "Evaluating the Scalability of a Big Data IoT Cloud Solution," Proc. of the 18th IEEE Int. Conf. on Smart Technologies (EUROCON 2019), pp. 1–5, Novi Sad, Serbia, 2019.

[29] U. Onoriode and G. Kotonya, "IoT Architectural Framework: Connection and Integration Framework for IoT Systems," First Workshop on Architectures, Languages and Paradigms for IoT EPTCS 264, Electronic Proceedings in Theoretical Computer Science, pp. 1-17, DOI: 10.4204/EPTCS.264.1, 2018.

[30] C. Modi, D. Patel, B. Borisaniya and H. Patel, "A Survey of Intrusion Detection Techniques in Cloud," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42–57, 2013.

[31] S. Yangui, R. H. Glitho, F. Belqasmi, M. J. Morrow and P. A. Polakos, "IoT End-user Applications Provisioning in the Cloud: State-of-the-Art," Proc. of IEEE International Conference on Cloud Engineering (IC2E), pp. 232-233, DOI: 10.1109/IC2E.2016.43, Berlin, Germany, 2016.

[32] M. Elkhodr, S. Shahrestani and H. Cheung, "The Internet of Things: New Interoperability, Management and Security Challenges," International Journal of Network Security & Its Applications, vol. 8, no. 2, pp. 85-102, 2016.

[33] P. Zdankin and T. Weis, "Longevity of Smart Homes," Proc. of IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 1-2, DOI: 10.1109/PerComWorkshops48775.2020.9156155, Austin, TX, USA, 2020.

[34] P. Sarwesh, N. S. V. Shet and K. Chandrasekaran, "Energy Efficient and Reliable Network Design to Improve Lifetime of Low Power IoT Networks," Proc. of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 117-122, DOI: 10.1109/WiSPNET.2017.8299731, Chennai, India, 2017.

[35] C. Butpheng, K.-H. Yeh and H. Xiong, "Security and Privacy in IoT-Cloud-Based e-Health Systems: A Comprehensive Review," Symmetry, vol. 12, no. 7, Article ID 1191, 2020.

[36] S. Aguzzi et al., "Definition of a Research and Innovation Policy Leveraging Cloud Computing and IoT Combination," Report by European Commission, [Online], Available: https://ec.europa.eu/digital-single-market/en/news/definition-research-and-innovation-policy-leveraging-cloud-computing-and-iot-combination, 2015.

[37] Y. Chen, "IoT, Cloud, Big Data and AI in Interdisciplinary Domains," Simulation Modeling Practice and Theory," vol. 102, 2020, DOI: 10.1016/j.simpat.2020.102070, 2020.

[38] R. Saboori, B. Sharma, N. Kumar and G. Gupta, "IoT-based Healthcare Support Services for Arrhythmia: A Review," Journal of Xi'an University of Architecture & Technology, vol. XII, no. VI, pp. 1035-1039, 2020.

[39] D. M. Donno, A. Giaretta, N. Dragoni, A. Bucchiarone and M. Mazzara, "Cyber-storms Come from Clouds: Security of Cloud Computing in the IoT Era," Future Internet, vol. 11, Article ID 127, DOI: 10.3390/fi11060127, 2019.

[40] A. Adamou et al., "Enabling Privacy and Security in Cloud of Things: Architecture, Applications, Security & Privacy Challenges," Applied Computing and Informatics, Elsevier, [Online], Available: https://doi.org/10.1016/j.aci.2019.11.005, 2019.

[41] A. Sajid, H. Abbas and K. Saleem, "Cloud-assisted IoT-based SCADA Systems Security: A Review of the State-of-the-Art and Future Challenges," IEEE Access, vol. 4, pp. 1375-1384, 2016.

[42] F. Md Sadek et al., "Threat Taxonomy for Cloud of Things," In Book: Internet of Things and Big Data Analysis: Recent Trends and Challenges, Edition 1, Chapter 5, pp. 149-190, United Scholars Publications, 2016.

[43] M. Babaghayou, N. Labraoui and A. A. A. Ari, "EPP: Extreme Points Privacy for Trips and Home Identification in Vehicular Social Networks," Proc. of the 3rd Edition of the National Study Day on Research on Computer Sciences (JERI2019), [Online], Available: http://ceur-ws.org/Vol-2351/paper_67.pdf, Saida, Algeria, 2019.

[44] R. Roman, J. Zhou and J. Lopez, "On the Features and Challenges of Security and Privacy in Distributed Internet of Things," Computer Networks, vol. 57, no. 10, pp. 2266-2279, 2013.

[45] S. Babar, A. Stango, N. Prasad, J. Sen and R. Prasad, "Proposed Embedded Security Framework for Internet of Things," Proc. of the 2nd IEEE International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), pp. 1–5, Chennai, India, 2011.

[46] C. Modi, D. Patel, B. Borisaniya et al., "A Survey on Security Issues and Solutions at Different Layers of Cloud Computing," Journal of Supercomputing, vol. 63, pp. 561–592, 2013.

[47] D. Catteddu and G. Hogben, "The European Network and Information Security Agency (ENISA): Emerging and Future Risk Programme." Computing, vol. 72, no. 1, pp. 2009–2013, 2009.

[48] Y. A. Hamza and M. D. Omar, "Cloud Computing Security : Abuse and Nefarious Use of Cloud Computing," Int. Journal of Computational Engineering Research, vol. 3, no. 6, pp. 22-27, 2013.

[49] Y. Zhang, D. Zheng, R.H. Deng, "Security and Privacy in Smart Health: Efficient Policy-hiding Attribute-based Access Control, " IEEE Internet of Things Journal, vol. 5, no. 3, pp. 2130-2145, 2018.

[50] Z. H. Hu, "The Research of Several Key Questions of Internet of Things," Proc. of IEEE International Conference on Intelligence Science and Information Engineering, pp. 362-365, Wuhan, China, 2011.

[51] H. Suo, J. Wan, C. Zou and J. Liu, "Security in the Internet of Things: A Review," Proc. of the International Conference on Computer Science and Electronics Engineering, pp. 648-651, DOI: 10.1109/ICCSEE.2012.373, Hangzhou, 2012.

[52] A. Alsaidi, "Security Attacks and Countermeasures on Cloud-assisted IoT Applications," Proc. of IEEE International Conference on Smart Cloud, pp. 213–217, NY, USA, 2018.

[53] K. N. Pallavi, V. R. Kumar and S. Srikrishna, "Comparative Study of Various Lightweight Cryptographic Algorithms for Data Security between IoT and Cloud," Proc. of the 5th International Conference on Communication and Electronics Systems (ICCES), pp. 589-593, DOI: 10.1109/ICCES48766.2020.9137984, Coimbatore, India, 2020.

[54] I. K. Dutta, B. Ghosh and M. Bayoumi, "Lightweight Cryptography for Internet of Insecure Things: A Survey," Proc. of the 9th IEEE Annual Computing and Communication Workshop and Conference (CCWC), pp. 0475-0481, DOI: 10.1109/CCWC.2019.8666557, Las Vegas, NV, USA, 2019.

[55] D. Kishore Kumar, G.Venkatewara Rao and G.Srinivasa Rao, "Cloud Computing: An Analysis of Its Challenges & Security Issues," Int. J. of Computer Science and Network (IJCSN), vol. 1, no. 5, 2012.

[56] X. Wang, Z. Ning, M. Zhou, X. Hu, L. Wang, B. Hu, R.Y. Kwok and Y. Guo, "A Privacy Preserving Message Forwarding Framework for Opportunistic Cloud of Things," IEEE Internet of Things Journal, vol. 5, no. 6, pp.5281–5295, 2018.

[57] S. Sharma, M. A. R. Shuman, A. Goel, A. Aggarwal, B. Gupta, S. Glickfield and I. D. Guedalia, "Context-aware Actions among Heterogeneous Internet of Things Devices," US Patent App., 14/187,156, 2014.

[58] A. Hameed and A. Alomary, "Security Issues in IoT: A Survey," Proc. of the International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), pp. 1-5, DOI: 10.1109/3ICT.2019.8910320, Sakhier, Bahrain, 2019.

[59] P. Eugster, S. Kumar, S. Savvides and J. J. Stephen, "Ensuring Confidentiality in the Cloud of Things," IEEE Pervasive Computing, vol. 18, no. 1, pp. 10-18, DOI: 10.1109/MPRV.2018.2877286, 2019.

[60] H. Zhu et al., "A Secure and Efficient Data Integrity Verification Scheme for Cloud-IoT Based on Short Signature," IEEE Access, vol. 7, pp. 90036–90044, 2019.

[61] F. Daneshgar, O. A. Sianaki and A. Ilyas, "Overcoming Data Security Challenges of Cloud of Things: An Architectural Perspective," Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2019), Part of the Advances in Intelligent Systems and Computing, vol. 993, Springer, Cham, DOI: 10.1007/978-3-030-22354-0_58, 2020.

[62] P. Eugster, S. Kumar, S. Savvides and J. Stephen, "Ensuring Confidentiality in the Cloud of Things," IEEE Pervasive Computing, vol. 18, pp. 10-18, DOI: 10.1109/MPRV.2018.2877286, 2019.

[63] K. Mwangi, M. Shedden and J. Mandu, "Modelling Malware Propagation on the Internet of Things Using an Agent-based Approach on Complex Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 1, pp. 26-40, 2019.

[64] Y. Hyunsik and K. Young, "Design and Implementation of High-availability Architecture for IoT-Cloud Services," Sensors, vol. 19, no. 15, Article ID 3276, 2019.

[65] M. A. Khan, S. Member and M. Y. Khan, "A Secure Framework for Authentication and Encryption Using Improved ECC for IoT-based Medical Sensor Data," IEEE Access, vol. 8, pp. 52018–52027, 2020.

[66] B. Adil and M. Ajaz, "Securing Communication in MQTT-enabled Internet of Things with Lightweight Security Protocol," EAI Endorsed Transactions on Internet of Things, vol. 3, no. 12, Article ID 154390, DOI: 10.4108/eai.6-4-2018.154390, 2018.

[67] B. Alohali, M. Merabti and K. Kifayat, "A Secure Scheme for a Smart House Based on Cloud of Things (CoT)," Proc. of the 6th IEEE Computer Science and Electronic Engineering Conference (CEEC), pp. 115–120, Colchester, UK, 2014.

[68] T. D. P. Bai and S. A. Rabara, "Design and Development of Integrated, Secured and Intelligent Architecture for Internet of Things and Cloud Computing," Proc. of the 3rd IEEE International Conference on Future Internet of Things and Cloud, pp. 817–822, Rome, Italy, 2015.

[69] S. Kumari, X. Li, F. Wu, A. K. Das, K.-K. R. Choo and J. Shen, "Design of a Provably Secure Biometrics-based Multi-cloud-server Authentication Scheme," Future Generation Computer Systems, vol. 68, pp. 320–330, 2017.

[70] A. Alrawais, A. Alhothaily, C. Hu and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," IEEE Internet Computing, vol. 21, no. 2, pp. 34–42, 2017.

[71] V. Valter, A. Aleksandar, P. Krešimir, M. Miljenko and Z. Ivana, "Adaptable Secure Communication for the Cloud of Things," Journal of Software: Practice and Experience, vol. 47, no. 3, pp. 489-501, 2017.

[72] K. Sundus and I. Almomani, "Mobility Effect on the Authenticity of Wireless Sensor Networks," Proc. of IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 286-292, DOI: 10.1109/JEEIT.2019.8717497, Amman, Jordan, 2019.

[73] B. Alohali, "Security in Cloud of Things (CoT)," In Book: Cloud Security: Concepts, Methodologies, Tools and Applications," IGI Global, pp. 1188–1212, DOI: 10.4018/978-1-5225-8176-5, 2019.

ملخص البحث:

في هذا العصر، الذي يوصف بعصر الاتصال والتشبيك، فإنّ إنترنت الأشياء تضيف الى المجالات التكنولوجية الموجودة وتجلب ثورة الى عالم تكنولوجيا المعلومات. وتتكون إنترنت الأشياء من أجهزة متصلة ببعضها البعض، قد تكون أجهزةً رقميةً أو ماديةً أو ميكانيكيةً مزودة بمحدِّدات فريدة ولها القدرة على إرسال المعلومات المحسوسة الى أجهزة أخرى بصورة مستقلة. وتُميَّز إنترنت الأشياء على أنها مؤلّفة من أجهزة مقيَّدة المصادر من حيث كفاءة المعالجة وسعة التخزين ومصادر الطاقة. ولمواكبة هذه المحدِّدات، يمكن استخدام التكنولوجيا القائمة المعروفة بالحوسبة السّحابية لتسهيل نظام إنترنت الأشياء عبر تخفيف متطلباته فيما يتعلق بالمعالجة والتخزين.

في هذه الورقة، تمت مناقشة ضرورة دمج السحابة وإنترنت الأشياء والفوائد المتأتّية من هذا الدمج. إضافة الى ذلك، تمّ تحديد عدة قضايا بحثية تنشأ من الدمج المشار إليه. من بين هذه القضايا، لوحظ أنّ المخاوف من غياب الأمن او الخصوصية هي من القضايا المحورية الناجمة عن دمج إنترنت الأشياء والحوسبة السّحابية، التي تحتاج الى معالجة لجعل الدمج ناجحاً. لقد جرى تحديد التهديدات المتعلقة بالأمن والخصوصية ، كما تم بحث آليّات الأمن القائمة في هذه الورقة.

من ناحية أخرى، تسلط الورقة الضوء على القضايا البحثية المفتوحة المرتبطة بالأمن والخصوصية في الأنظمة التي تقوم على دمج إنترنت الأشياء والحوسبة السحابية. ويمكن لهذه الورقة أن تكون أساساً للبحث الذي تبرز الحاجة اليه في مجال قضايا الأمن والخصوصية في مثل هذه الأنظمة (سحابة الأشياء CoT).

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

IMPROVED DEEP LEARNING ARCHITECTURE FOR DEPTH ESTIMATION FROM SINGLE IMAGE

Suhaila F. A. Abuowaida and Huah Yong Chan

(Received: 28-Jun.-2020, Revised: 3-Aug.-2020 and 20-Sep.-2020, Accepted: 27-Sep.-2020)

ABSTRACT

Numerous benefits of depth estimation from the single image field on medicine, robot video games and 3D reality applications have garnered attention in recent years. Closely related to the third dimension of depth, this operation can be accomplished using human vision, though considered challenging due to the various issues when using computer vision. The differences in the geometry, the texture of the scene, the occlusion scene boundaries and the inherent ambiguity exist because of the minimal information that could be gathered from a single image. This paper, therefore, proposes a novel depth estimation in the field of architecture, which includes the stages that can manage depth estimation from a single RGB image. An encoder-decoder architecture has been proposed, based on the improvement yielded from DenseNet that extracted the map of an image using skip connection technique. This paper also takes on the reverse Huber loss function that essentially suits our architecture hand driven by the value distributions that are commonly present in depth maps. Experimental results have indicated that the depth estimation architecture that employs the NYU Depth v2 dataset has a better performance than the other state-of-the-art methods that tend to have fewer parameters and require fewer training time.

KEYWORDS

Depth estimation, Single image, Deep learning, Encoder-decoder.

1. INTRODUCTION

Numerous benefits in the use of depth estimation from the single image field on medicine [33], robot video games [4] and 3D reality applications [15] have resulted in a spike of interest to the operations of 3D in recent years. Closely related to the third dimension of depth, this application can be accomplished using human vision, but is considered challenging through the computer vision. However, the differences in the geometry, the texture of a scene, the occlusion scene boundaries and the inherent ambiguity could not be captured in depth when employing a single image [26]. The estimation of depth is essential to obtain the ideal distance for each pixel in a single image between an observer and the visual detail [15]. Traditional algorithms that manage monocular images in recognising the dimension of depth, which include Structure from Motion (SfM) [27], as well as differences in shading and lighting of images [28] [1], require specific environmental assumptions. Besides, there are other issues related to finding the depth of the images, such as determining the heterogeneity of the depth [3] and the quality image processing after identifying the depth [22]. Initially, researchers have focused on the stereo vision to acquire depth estimation by using multi-view images [20] [32] [21]. Nevertheless, this method appears to have several setbacks, such as low efficiency of depth estimation due to blind region repetition and unmatching texture at areas of the same point. Since then, the use of a single image for depth estimation has been given much attention [7] [18] [9] due to the manageable cost, specialized equipment in use and flexibility in capturing the image. The devices, though light in weight, are reliable. By adapting Markov Random Field (MRF) [24] and Conditional Random Field (CRF) [19] [16] [29], the first algorithms; superpixels, are created to be used in discovering the depth from a single image.

Recently, deep learning has been used in computer vision tasks and proven to be useful in obtaining satisfactory results. The Convolutional Neural Network (CNN) of deep learning was receiving much attention for handling computer vision applications, such as object recognition [11] [13] [30] and segmentation [6] [17] [12], due to the self-learning feature. A study [7] has proposed a framework that would be the first to integrate CNN for depth estimation through multi-scale CNN. However, the framework took a long time to produce a depth estimation for each image. Since then, many methods

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

based on CNN had been proposed, as reported in a study by Eigen et al. [7]. Several researchers [16] [31] suggested a framework that merged CNN and CRF methods. CNN would extract features from the image and CRF would provide the final result of the prediction. However, these algorithms had high computation time and issues with the inferences.

The process of creating a CNN that contained multiple layers was also complicated due to the emergence of gradient vanishing problem in the training process. Many studies were affected by the increase in the number of these layers in CNN, which demonstrated a positive effect of obtaining high performance [14] [10] [2] [8]. The ResNet architecture designed by He et al. [11] was adopted in a study by Laina et al. [14] to observe up-sampling blocks for depth estimation, which found that the property related to translating invariance that existed in deep CNN may adversely affect the process of depth estimation. This issue, however, could be overcome by a skip connection technique [5] [2] [23]. The researcher in [2] utilizes the DenseNet [13] to determine the depth of a single image. Despite that, this algorithm was needing a long computation time, due to adopting DenseNet [13].

Hence, this paper intends to use a novel depth estimation architecture that can discern depth estimation from a single RGB image to improve the accuracy of depth estimation and reduce the number of parameters, which affects reduced computation time and is efficient even with the use of a huge dataset. This architecture consists of many stages, whereby firstly an encoder-decoder architecture has been improved from DenseNet [13] to deal with some implied problems in DenseNet that still exist, such as recognition of layers that have failed to have enough training, recognition of layers that have had more focused training and the use of a large filter size for the first convolution layer. Then, the standard loss function can be observed and adopted. The evaluation of depth estimation tends to be carried out by employing four types of measurements, which are average relative error (rel), root mean squared error (rms), average (log10) error and threshold accuracy. The algorithms in the proposed architecture of this study were also compared to algorithms used in other studies, such as Eigen et al. [7], Laina et al. [14], Alhashim et al. [2], Hao et al. [10], Wang et al. [29], Ren et al. [23] and Carvalho et al. [5]. The results from the four types of algorithms in measurement are described before concluding the observations of this study.

2. METHODOLOGY

This paper deals with investigating depth estimation of a single RGB image using an end-to-end learning architecture that produces a direct mapping of RGB in depth, as shown in Figure 1.

2.1 Encoder-Decoder Architecture

Figure 1 shows the proposed encoder-decoder architecture for depth estimation from a single RGB image. Many researchers have argued that the performance of the CNN architecture may increase with the depth of the CNN architecture. Nevertheless, stacking many layers on the CNN architecture cannot guarantee improved performance of the network and may alternatively lead to a significant decrease in performance. This issue exists because of the gradient vanishing problem during the training phase [24], which happens when the CNN architecture is stacked with too many layers. By using the DenseNet, the vanishing problem can be avoided through a connection between the layers. However, the DenseNet has been found to disregard the activation layer during the backpropagation process, as there is no formula within the parameters of DenseNet that describes the changing process, which leads to reduced accuracy in the gradient formula. The formula used in the DenseNet may not ascertain the layers that need more training than others. The novel architecture proposed in this study has improved the DenseNet [13], as shown in Figure 2, by simplifying and analyzing forward and backward propagation. The new rules of the different parameters in the DenseNet are obtained based on the new gradient formula in determining the layer that needs more or reduced training. A filter size more suitable than DenseNet is also selected to extract high and low levels of the features from the input image and the reduction parameters requirement, which leads to a reduced computation time based on the Formula 2.

- **DenseNet Analysis**

The connection between the layers through the gradient formula [13] is the key to solve the gradient vanishing problem. However, there are challenges when directly inferring forward and backward

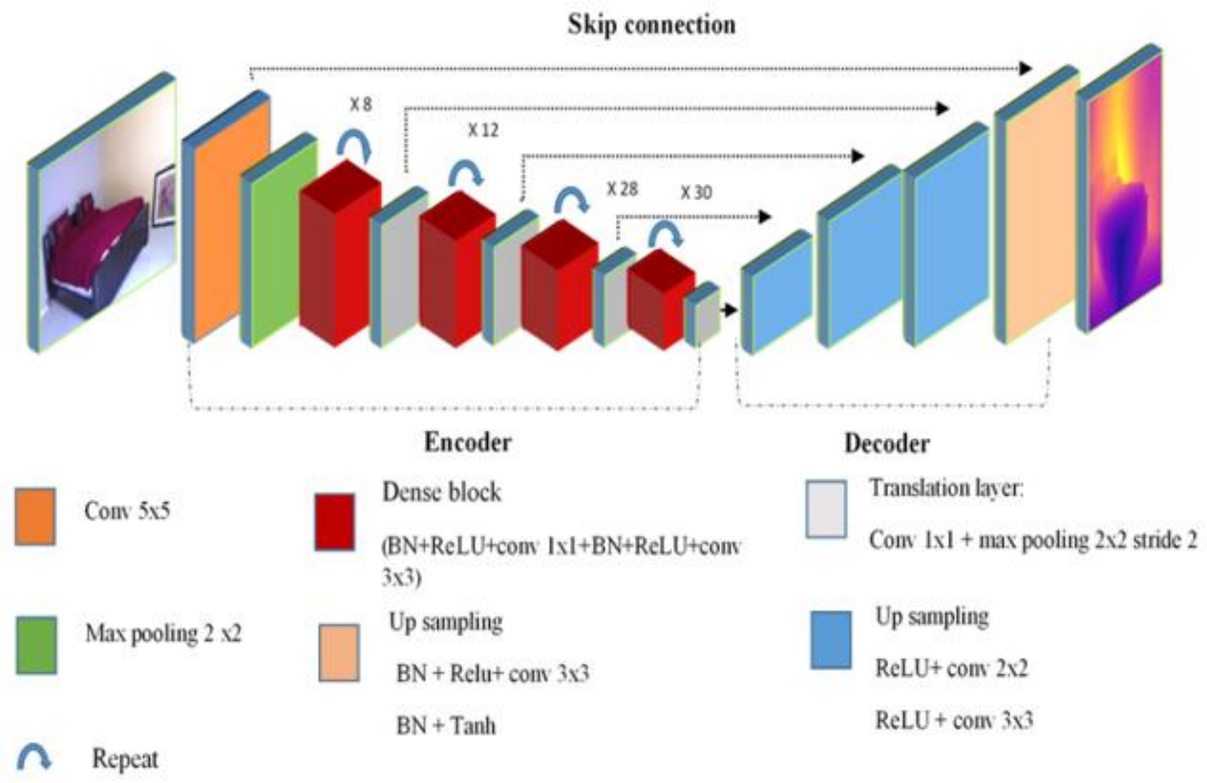


Figure 1. Encoder-decoder architecture for depth estimation from a single RGB image.

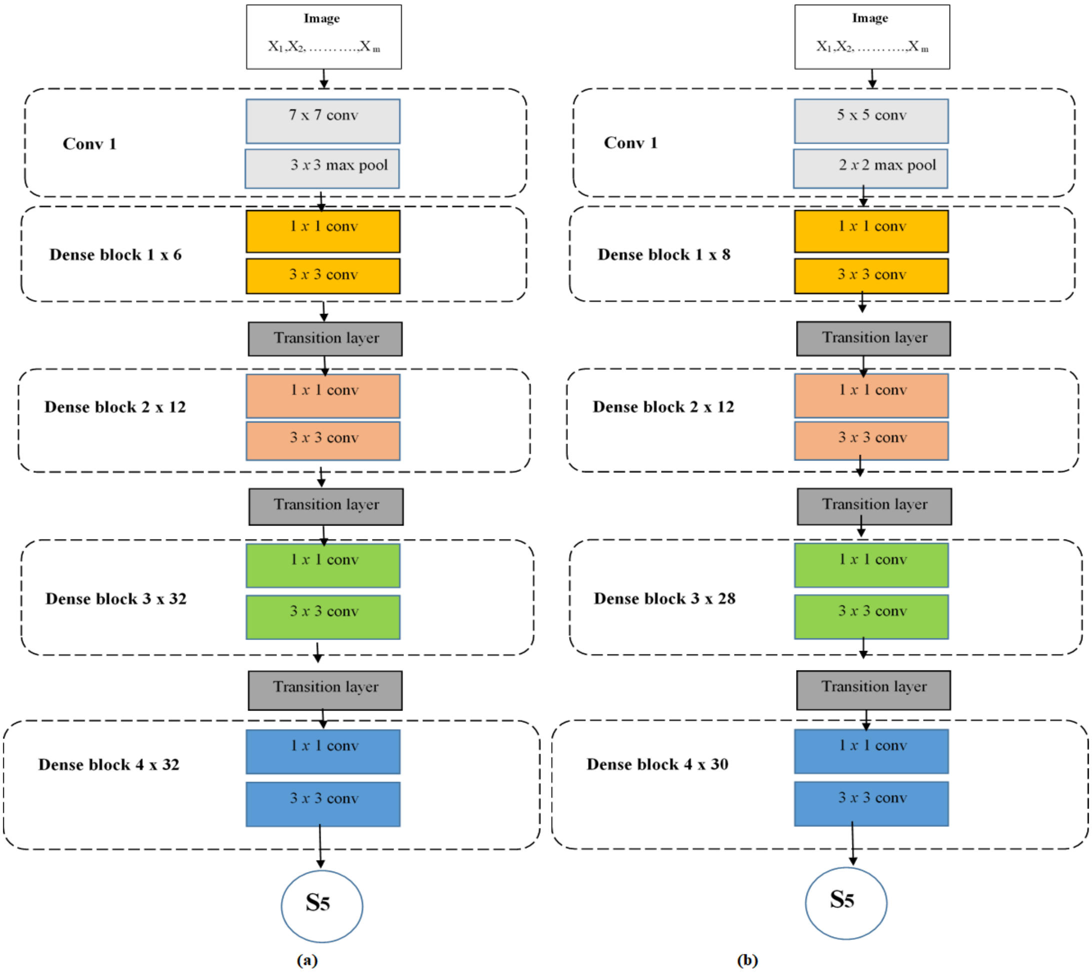


Figure 2. (a) DenseNet architecture 169 and (b) Improved DenseNet architecture.

propagation of DenseNet through gradient formula. Therefore, forward propagation and backward propagation of the DenseNet within a network addressing gradient vanishing are analyzed.

- **Forward Propagation Analysis**

The total loss function in the DenseNet is calculated using the square of the difference between the predicted output and the ground truth, as represented in the following formula:

$$L = \frac{1}{N}\sum_{i=1}^{c} |\,\hat{y}_i - y_i|^2 = \frac{1}{2}\sum_{i=1}^{c} |\,e_i|^2 \tag{1}$$

where N = The normalization term, y_i= The ground truth value, $\hat{y}_i$ = Predicted value, e_i= $y_i - \hat{y}_I$ and c = Number of classification layers.

Forward propagation, which is the first convolution layer the DenseNet, is represented by the following formula:

$$s_0 = \sum_{i=0}^{n} x_i . w_i \tag{2}$$

The dense block (DenseB) contains the h(.) function that has three operations: Batch normalization (BN), Rule layer and convolution kernel, which is a set of the first input layer [s_0, s_1, ..., s_{i-1}], where each layer receives the maps of the feature from all previous layers as input, as demonstrated by Formula 3.

$$DenseB = w_1h_1(s_0) + w_2h_2(s_0, s_1) + w_3h_3(s_0, s_1, s_2) + w_4h_4(s_0, s_1, s_2, s_3) \tag{3}$$

$$DenseB_i = \sum_{\mathrm{j}=1}^{R}\sum_{i=1}^{4} w_i\, h_i(s) \tag{4}$$

where R is represents the repeated number of the dense block.
Then, the feature maps from the dense block input to the transaction layer are connected to the different dimensions through the following formula:

$$y_0 = \sum_{i=0}^{n} w_i\ .\theta(DenseB_i) \tag{5}$$

where θ(.) is the activation function and y_0 is the output from the first transaction layer. Forward propagation for the encoder uses the following formula:

$$y_j = \sum_{\mathrm{j}=0}^{3}\sum_{i=1}^{N} w_i\,.\,\theta(DenseB_i) \tag{6}$$

where j= 1...3 (number of dense block in the architecture).

- **Back Propagation Analysis**

The predicted output of the simplified encoder is obtained from the weight of the last layer that employed backpropagation. The gradient, L, is represented in the following formulae.

$$\frac{\partial L}{\partial \omega_B} = \frac{\partial L}{\partial \hat{y}_B}\frac{\partial \hat{y}_B}{\partial \omega_B}$$

$$= \delta_B\,\frac{\partial\left(\Sigma_{i=1}^{4}\sum_{j=1}^{N} W_{B.}\theta(S_{Bi})\right)}{\partial \omega_B} = \delta_B.\,\theta(S_{Bi}) \tag{7}$$

where

$$\delta_{B_4} = \frac{\partial L}{\partial S_{B_4}}$$

$$= \frac{\partial \frac{1}{m} \Sigma_{\mathrm{i}=1}^{c} (\hat{y}_i - y_i)}{\partial S_{B_4}}$$

$$= e_i . \theta'(S_{B_4}) \tag{8}$$

The hidden layers of the encoder of this study have the same gradient formula:

$$\delta_i = \theta' \left(y_i . \sum_{i=0}^{3} \delta_{i+1} . w_{i+1} \right) \tag{9}$$

The gradient of the first connected weight fades as the number of layers increases in the network. This study defines $\Delta\delta^n$ as the gradient that increases based on the number of layers (n) in the encoder, as shown in the following formula.

$$\triangle\, \delta_i^n = \theta'(y_i^n) \sum_{i=0}^{c} \delta_i^{n+1}, \qquad n = 1, \dots, 4 \tag{10}$$

$\triangle\, \delta_i^n$ solved the vanishing problem in a deep network.

Skip connection technique is used to connect the encoder and decoder transferring the features of the maps to the decoder during the up-sampling process for depth estimation, which tends to speed up the learning of context awareness and overcome the translation invariance. The decoder in this study uses bi-linear up-sampling, as shown in Figure 1, where the up-sampling block utilizes ReLU for activation and convolution of the layers.

2.2 Loss Function

In this study, the encoder-decoder architecture has adopted various loss functions as represented in the following formulae:

$$L_{meanabsolute}(L_1) = \frac{1}{N} \sum_{i=1}^{s} | \hat{y}_i - y_i | \tag{11}$$

$$L_{meansquare}(L_2) = \frac{1}{N} \sum_{i=1}^{s} (\hat{y}_i - y_i)^2 \tag{12}$$

$$L_{huber} = \begin{cases} L_1(l_i) & L_1(l_i) \geq c, \\ \dfrac{L_2(l_i) + c^2}{2c} & else \end{cases} \tag{13}$$

$$L_{berhub} = \begin{cases} L_1(l_i) & L_1(l_i) \leq c, \\ \dfrac{L_2(l_i) + c^2}{2c} & else \end{cases} \tag{14}$$

where yi= The ground truth value, yˆi = Predicted value, s = Number of classification layers, N = Normalization term, $c = \frac{1}{5}\max(|\hat{y}_i - y_i|)$ and i = Index value of pixel for each depth image in the current batch.

3. Results and Discussion

The experimental results and evaluation of the algorithms presented have been analysed and interpreted. The evaluation of the encoder-decoder architecture has employed the following four measurements:

1) Average relative error (rel) =

$$\frac{1}{n} \sum_{p}^{n} \frac{|y_i - \hat{y}_i|}{y} \tag{15}$$

2) Root mean squared error (rms) =

$$\sqrt{\frac{1}{n}\sum_{p}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

3) Average ($\log_{10}$) error =

$$\frac{1}{n}\sum_{p}^{n} |\log_{10} y_i - \log_{10} \hat{y}_i| \tag{17}$$

4) Threshold accuracy =

$$\max\left(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i}\right) = \delta < threshold = 125{,}125^2, 125^3 \tag{18}$$

where y_i= The ground truth value, $\hat{y}_i$ = Predicted value and n= Total value of pixel for each depth image.

3.1 Experimental Specifications

The architecture implemented in this study is built using TensorFlow [11], Amazon Web Services (AWS) and Amazon Machine Image (AMI), whereby the GPU-Us-Tesla V100 was at 16GB, while the VCPUs-8 cores had 61GB.

The algorithm of optimization used in is Stochastic Gradient Descent (SGD) [13]. The weight decay is 0.0001, with the learning momentum at 0.9 and the learning rate at 0.001 for 20 epochs.

3.2 Dataset

The quality of depth estimation has been evaluated using the NYU Depth v2 benchmark [25], which is considered one of the most well-known datasets for RGB single-image depth estimation. This dataset contains 1449 densely labeled pairs of images from indoor scenes with depth, 464 new scenes and 407,024 new unlabeled images that have been captured using Microsoft Kinect. Based on previous works that employed the NYU Depth v2 benchmark in examining depth estimation [7] [14] [2], the standard training and testing split was used to evaluate 654 image-depth pairs from the set.

3.3 Backbone Result

The proposed network has been chosen through the specific number of duplicates for each dense block based on the acquired results, as shown in Table 1. A selection of suitable duplicates for each layer of the dense block is carried out to improve the performance of the backbone network after the filter size is decreased.

Table 1. Number of duplicates for each dense block.

Block of dense layers	Number of duplicates	↓ *rel*
1	**1**,1,1,1,1	0.6630
	2,1,1,1,1	0.6656
	3,1,1,1,1	0.6705
2	1,2,1,1,1	0.5410
	1,4,1,1,1	0.5403
	1,6,1,1,1	0.5333
	1,**8**,1,1,1	0.5200
	1,10,1,1,1	0.5170
3	1,8,2,1,1	0.4801
	1,8,4,1,1	0.4711
	1,8,6,1,1	0.4287
	1,8,8,1,1	0.4122
	1,8,**1**2,1,1	0.4036
	1,8,12,2,1	0.3885

4	1,8,12,4,1	0.3806
	1,8,12,6,1	0.3705
	1,8,12,8,1	0.3674
	1,8,12,10,1	0.3506
	1,8,12,12,1	0.3302
	1,8,12,14,1	0.3089
	1,8,12,16,1	0.2883
	1,8,12,18,1	0.2615
	1,8,12,20,1	0.2505
	1,8,12,22,1	0.2366
	1,8,12,24,1	0.2307
	1,8,12,26,1	0.2277
	1,8,12,**28**,1	0.2186
	1,8,12,30,1	0.2185
5	1,8,12,28,2	0.2092
	1,8,12,28,4	0.2025
	1,8,12,28,6	0.2003
	1,8,12,28,8	0.1945
	1,8,12,28,12	0.1920
	1,8,12,28,14	0.1858
	1,8,12,28,16	0.1789
	1,8,12,28,18	0.1603
	1,8,12,28,20	0.1552
	1,8,12,28,22	0.1501
	1,8,12,28,24	0.1483
	1,8,12,28,26	0.1382
	1,8,12,28,28	0.1241
	1,8,1228,,**30**	0.1220
	1,8,12,28,32	0.1220

Based on Table 1, the most suitable duplicates for each dense block are as follows:

1) One-time repetition of the first convolution layer.
2) Eight-time repetition of the second dense block.
3) Repetition of the third dense block 12 times.
4) Repetition of the fourth dense block 28 times.
5) Repetition of the fifth dense block 30 times.

As shown in Table 1, it is found that some dense blocks need to be repeated more to obtain better results (second dense block), while some dense blocks do not need duplicates of training. So we have reduced the number of repetition of training for these dense blocks (fourth dense block and fifth dense block), because it not needed to repeat the training, which leads to reduce parameters and then leads to reduce computation time.

3.4 Loss Function

The various loss functions of the encoder-decoder architecture tend to be compared using mean absolute, mean square, Huber and BerHub. Table 2 shows the results of this comparison.

As shown in Table 2, the performance of the loss function using BerHub is found to be the best for the different measurements used, which are rel error, rms error, $\log_{10}$ error, $\theta < 1.25$ accuracy, $\theta < 1.25^2$ accuracy and $\theta < 1.25^3$ accuracy. The results for the architecture are 0.1220, 0.4584, 0.0531, 0.8525, 0.9735 and 0.9946, respectively per sequence of measurements, as previously mentioned.

The loss function using Berhub is also found to be balanced between ground truth depth map values. When the differences between the ground truth depth map values are small, there will not be big differences in the weights and the dependence in this case is on L_1, but when the differences between

Table 2. Performance of various loss functions.

Loss function	↓ *rel*	↓ *rms*	↓ $\log_{10}$	↑ $\theta < 1.25$	↑ $\theta < 1.25^2$	↑ $\theta < 1.25^3$
Mean absolute	0.1367	0.5809	0.0582	0.8214	0.9670	0.9926
Mean square	0.1308	0.4820	0.0557	0.8408	0.9708	0.9934
Huber	0.1357	0.4949	0.0563	0.8363	0.9724	0.9942
BerHub	**0.1220**	**0.4584**	**0.0531**	**0.8525**	**0.9735**	**0.9946**

the ground truth depth map values are large, we tend to choose the equation $\frac{L_2(l_i)+c^2}{2c}$ and this equation is meant to reduce the loss in weights, which leads to a convergence between $(\hat{y}_i, y_i)$ so that we can get the best distribution of the ground truth depth map values. Hence, the loss function using Berhub is also found to be more appropriate for the architecture proposed in this study because of the small residuals that are utilized in the training stage, which has resulted in a better weight adjustment to achieve a better result.

3.5 Encoder-Decoder Comparison with Various Backbones

Table 3 presents the results of a comparison between various implementations to the backbone.

Table 3. Performance of various implementations of backbone.

Backbone	↓ *rel*	↓ *rms*	↓ $\log_{10}$	Parameters	Computation time per hour
DenseNet 121	0.1312	0.4970	0.0571	**21.2M**	**16.28**
DenseNet 169	0.1281	0.4740	0.0551	47.0M	21.13
DenseNet 201	0.1289	0.5515	0.0537	55.9M	26.34
ResNet50	0.1571	0.5590	0.0672	49.5M	30.55
ResNet101	0.1441	0.5559	0.0687	68.5M	40.23
Ours	**0.1220**	**0.4584**	**0.0531**	44.3M	19.31

The performance of the proposed backbone was found to exhibit better results. The rel error, rms error and $\log_{10}$ error accuracy of our backbone network amounted to 0.1220, 0.4584 and 0.0531. These values were significantly higher compared to other backbone values. The proposed backbone network solved the issue of vanishing gradient through identity shortcut based on a new gradient formula while taking the efficiency training for each dense block into account, as shown in Table 1. This aspect was considered through a specific duplicate increase or decrease in training and the reduction in the filter size to extract a certain amount of features from the input image and transfer this amount to other layers. The benefit obtained from the features led to positive results compared to other algorithms. The proposed backbone has consumed 44.3M parameters and 19.31 computation time per hour, making it the second in terms of the parameters and computation time per hour after DenseNet 121 backbone due to the increased repetition for some dense blocks to obtain better results, despite the high-speed characterization of the DenseNet 121 backbone algorithms in this process. Simultaneously, this is due to accuracy in terms of rel error, rms error and $\log_{10}$ error.

3.6 Comparison with the State of the Art Architecture

The results from this study are compared to those of studies conducted by Eigen et al. [7], Laina et al. [14], Al- hashim et al. [2], Hao et al. [10], Wang et al. [29], Ren et al. [23] and Carvalho et al. [5]. Table 4 shows the results of these comparisons. As shown in Table 4, the performance of the proposed architecture in this study obtained better results compared to other types of architectures when calculated using the different measurements; rms error, $\log_{10}$ error, $\theta < 1.25$ accuracy, $\theta < 1.25^2$ accuracy and $\theta < 1.25^3$ accuracy. The results of the proposed architecture are 0.4584, 0.0531, 0.8525, 0.9946, respectively, per the sequence of measurement previously mentioned. This result has suggested that the framework of the new backbone proposed in this study is extremely crucial in achieving desirable results. The proposed network has taken into account the efficiency of training for each dense block by increasing specific duplicates or decreasing training. Moreover, the filter size is decreased to obtain the number of features extracted from the input image so as to be fed to other layers. These steps have resulted in a better performance compared to other algorithms. Besides, this study has adapted the various loss functions and chosen the most suitable loss function for the proposed architecture, as shown in Table 2. The visual experimental results obtained from this study are clearly shown in Figure 3. The visuals illustrated in sequence are: the original RGB image, the depth prediction and the map of ground truth

Table 4. Performance of state-of-the-art architectures on NYU Depth v2 dataset.

Architecture	↓ *rel*	↓ *rms*	↓ $\log_{10}$	↑ $\theta < 1.25$	↑ $\theta < 1.25^2$	↑ $\theta < 1.25^3$
Eigen et al.[7]	0.158	0.641	-	0.769	0.950	0.988
Laina et al. [14]	0.127	0.573	0.055	0.811	0.953	0.988
Alhashim et al.[2]	0.123	0.465	0.053	0.846	**0.974**	0.994
Hao et al. [10]	0.127	0.555	0.053	0.841	0.966	0.991
Wang et al. [29]	0.220	0.745	0.094	0.605	0.890	0.970
Carvalho et al. [5]	0.135	0.600	0.059	0.819	0.957	0.987
Ren et al. [23]	**0.113**	0.501	-	0.833	0.968	0.993
Ours	0.122	**0.458**	**0.052**	**0.853**	**0.974**	**0.995**

.

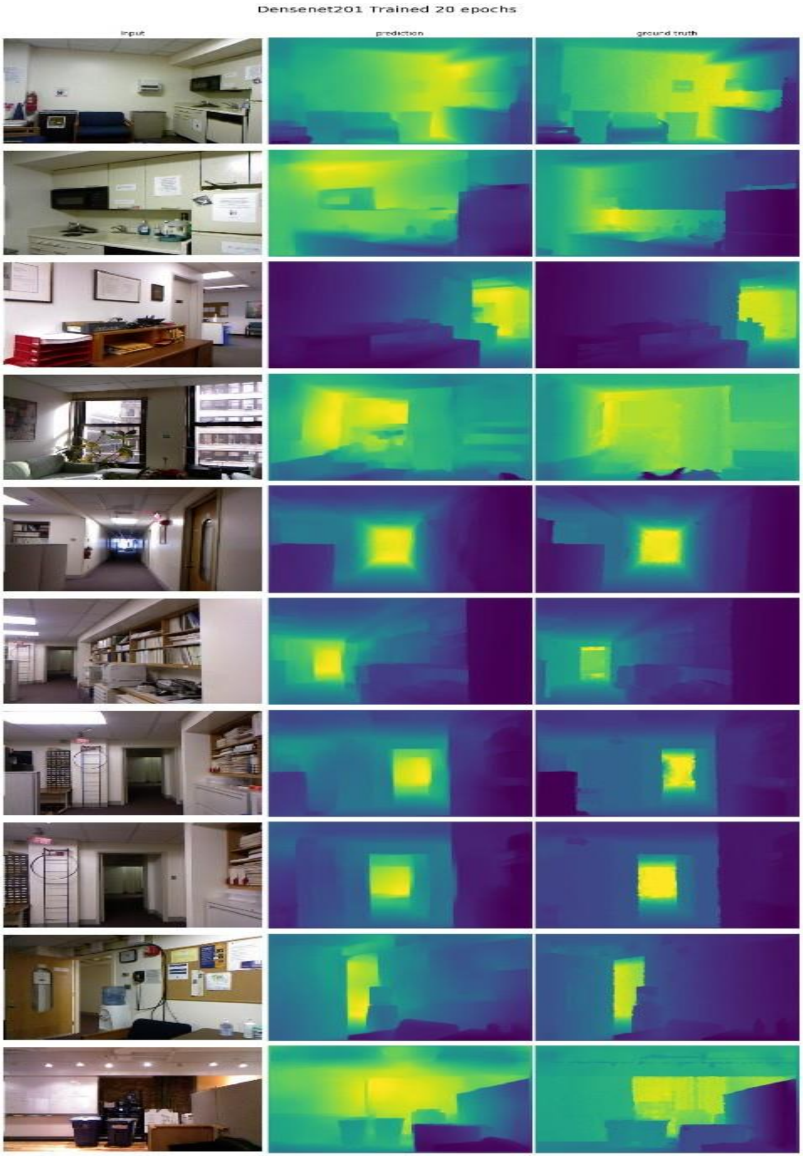


Figure 3. The visual experimental results of our encoder- decoder architecture from NYU Depth v2 dataset.

4. CONCLUSION

In this study, a novel encoder-decoder architecture has been proposed to investigate depth estimation from a single RGB image. There are several stages in this architecture, whereby firstly the encoder-decoder architecture has been simplified from the DenseNet and analyzed using forward and backward propagation. Then, the new rules for the different parameters of the DenseNet are obtained based on the new gradient formula to search for the layer that needs more or less training. The filter size that is selected to extract high and low feature levels from the input image for solving the gradient vanishing problem should be more suitable than DenseNet. Results from this study on the NYU Depth v2 dataset have also shown that the loss function using Berhub has produced the best performance. Experiments on the NYU Depth v2 dataset further demonstrated that the encoder-decoder architecture in this study has achieved a state-of- the-art performance based on the consistent performance in obtaining depth estimation from a single RGB image.

REFERENCES

[1] A. Abrams, C. Hawley and R. Pless, "Heliometric Stereo: Shape from Sun Position," European Conference on Computer Vision, Part of the Lecture Notes in Computer Science Book Series (LNCS), Vol. 7573, pp. 357–370, Springer, 2012.

[2] I. Alhashim and P. Wonka, "High Quality Monocular Depth Estimation *via* Transfer Learning," arXiv: 1812.11941v2, [Online], Available: https://arxiv.org/pdf/1812.11941.pdf, 2018.

[3] A. Atapour-Abarghouei and T. P. Breckon, "Real-time Monocular Depth Estimation Using Synthetic Data with Domain Adaptation *via* Image Style Transfer," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE/CVF), pp. 2800–2810, Salt Lake City, UT, USA, 2018.

[4] T. Bebie and H. Bieri, "A Video-based 3D-Reconstruction of Soccer Games," Computer Graphics Forum, vol. 19, no. 3, pp. 391–400, DOI: 10.1111/1467-8659.00431, 2000.

[5] M. Carvalho et al., "On Regression Losses for Deep Depth Estimation," Proc. of the 25th IEEE International Conference on Image Processing (ICIP), pp. 2915–2919, Athens, Greece, 2018.

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

[7] D. Eigen, C. Puhrsch and R. Fergus, "Depth Map Prediction from a Single Image Using a Multi-scale Deep Network," Advances in Neural Information Processing Systems, arXiv: 1406.2283v1, pp. 2366–2374, [Online], Available: https://arxiv.org/pdf/1406.2283.pdf, 2014.

[8] H. Fu et al., "Deep Ordinal Regression Network for Monocular Depth Estimation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE/CVF), pp. 2002–2011, Salt Lake City, UT, USA, 2018.

[9] A. Grigorev et al., "Depth Estimation from Single Monocular Images Using Deep Hybrid Network," Multimedia Tools and Applications, vol. 76, no. 18, pp. 18585–18604, 2017.

[10] Z. Hao et al., "Detail Preserving Depth Estimation from a Single Image Using Attention Guided Networks," Proc. of the IEEE International Conference on 3D Vision (3DV), pp. 304–313, Verona, Italy, 2018.

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

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

[13] G. Huang et al., "Densely Connected Convolutional Networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708, Honolulu, HI, USA, 2017.

[14] I. Laina et al., "Deeper Depth Prediction with Fully Convolutional Residual Networks," Proc. of the 4th IEEE International Conference on 3D Vision (3DV), pp. 239–248, Stanford, CA, USA, 2016.

[15] W. Lee, N. Park and W. Woo, "Depth-assisted Real-time 3D Object Detection for Augmented Reality," Proc. of the 21st International Conference on Artificial Reality and Telexistence, (ICAT), vol. 11, no. 2, pp. 126–132, Osaka, Japan, 2011.

[16] B. Li et al., "Depth and Surface Normal Estimation from Monocular Images Using Regression on Deep Features and Hierarchical CRFs," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVRP), pp. 1119–1127, Boston, MA, USA, 2015.

[17] Y. Li et al., "Fully Convolutional Instance-aware Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVRP), pp. 2359–2367, Honolulu, HI, USA, 2017.

[18] F. Liu et al., "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pp. 2024–2039, 2015.

[19] M. Liu, M. Salzmann and X. He, "Discrete-continuous Depth Estimation from a Single Image," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 716–723, Columbus, OH, USA, 2014.

[20] Y. Liu et al., "Continuous Depth Estimation for Multi-view Stereo," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2121–2128, Miami, FL, USA, 2009.

[21] P. K. Martin et al., "Improved Depth Map Estimation from Stereo Images Based on Hybrid Method," RadioEngineering Journal, vol. 21, no. 1, pp. 70-78, 2012.

[22] F. Qi et al., "Structure Guided Fusion for Depth Map Inpainting," Pattern Recognition Letters, vol. 34, no. 1, pp. 70–76, 2013.

[23] H. Ren, M. El-Khamy and J. Lee, "Deep Robust Single Image Depth Estimation Neural Network Using Scene Understanding," Computer Vision and Pattern Recognition Workshops, arXiv: 1906.03279v1, [Online], Available: https://arxiv.org/pdf/1906.03279.pdf, pp. 37–45, 2019.

[24] A. Saxena, M. Sun and A. Y. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 824–840, 2008.

[25] N. Silberman et al., "Indoor Segmentation and Support Inference from RGBD Images," Proc. of European Conference on Computer Vision, Part of the Lecture Notes in Computer Science Book Series (LNCS), vol. 7576, pp. 746–760, Springer, 2012.

[26] F. Simões et al., "Challenges in 3D Reconstruction from Images for Difficult Large-scale Objects: A Study on the Modeling of Electrical Substations," Proc. of the 14th IEEE Symposium on Virtual and Augmented Reality, pp. 74–83, Rio de Janiero, Brazil, 2012.

[27] R. Szeliski, Computer Vision: Algorithms and Applications (Texts in Computer Science), 2011 Edition, Springer, 2011.

[28] M. W. Tao et al., "Depth from Shading, Defocus and Correspondence Using Light-field Angular Coherence," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1940–1948, Boston, MA, USA, 2015.

[29] P. Wang et al., "Towards Unified Depth and Semantic Prediction from a Single Image," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2800–2809, Boston, MA, USA, 2015.

[30] Y. C. Wong et al., "Deep Learning-based Racing Bib Number Detection and Recognition," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 181-194, 2019.

[31] D. Xu et al., "Multi-scale Continuous CRFs As Sequential Deep Networks for Monocular Depth Estimation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5354–5362, Honolulu, HI, USA, 2017.

[32] H. Xu, Y. Cai and R. Wang, "Depth Estimation in Multi-view Stereo Based on Image Pyramid," Proceedings of the 2nd International Conference on Computer Science and Artificial Intelligence, pp. 345–349, [Online], Available: https://doi.org/10.1145/3297156.3297238, 2018.

[33] S. Zachow, M. Zilske and H.-C. Hege, "3D Reconstruction of Individual Anatomy from Medical Image Data: Segmentation and Geometry Processing," Proc. of the 25th ANSYS Conference & CADFEM Users' Meeting, Proc. CD 2.12.15, ZIB-Report, pp. 7-41, ISSN: 1438-0064, Dresden, Germany, 2007.

ملخص البحث:

جلبت الفوائد العديدة لتقدير العُمق من مجال صورةٍ واحدةٍ في الطّبّ وألعاب الفيديو الخاصة بالروبوت وتطبيقات الواقع ثلاثية الأبعاد الكثير من الاهتمام في السنوات الأخيرة. ولعلاقتها الوثيقة بالبعد الثالث المتمثل في العُمق، يمكن إنجاز هذه العملية باستخدام رؤية الإنسان؛ نظراً لاعتبارها محفوفةً بالتحديات بسبب قضايا متنوعة عند استخدام رؤية الحاسوب. فالاختلافات في الهندسة، ونسيج المشهد، والحدود الخاصة بانسداد المشهد، والغموض المتأصّل؛ كلها قضايا موجودة بفعْل ضآلة المعلومات التي يمكن الحصول عليها من صورةٍ مفردةٍ.

لذلك، فإنّ هذه الورقة تقترح طريقةً مبتكرةً لتقدير العُمق في مجال المعمارية، تتضمن المراحل التي يمكنها أن تقوم بتقدير العُمق من صورةٍ مفردةٍ بالأحمر والأخضر والأزرق (RGB). ولقد تمّ اقتراح معمارية مؤلفة من وحدة ترميز وأخرى لفكّ الترميز، بناءً على التحسين الذي تم الحصول عليه من (DenseNet)؛ إذ جرى استخلاص خريطة للصورة باستخدام تقنية التوصيل القائمة على التّخطّي. ومن ناحيةٍ أخرى، تعتمد هذه الورقة دالّة الفقْد العكسية لهوبر (Huber) التي تلائم بصورة أساسية المعمارية المقترحة من خلال توزيعات القيم التي تُوجد بشكلٍ عام في خرائط العُمق.

وقد أشارت النتائج التجريبية الى أنّ معمارية تقدير العُمق التي توظِّف مجموعة البيانات (NYU Depth v2) كانت ذات أداءٍ أفضل مقارنة بالطرق الأخرى المستخدمة في مجال تقدير العُمق التي تميل الى امتلاك عددٍ أقلّ من المتغيرات وتتطلب زمن تدريبٍ أقصر.

JJCIT Annual List of Reviewers (2020)

Name, *Affiliation*, Country

Zaher Mundher Yaseen, *Ton Duc Thang University*, Vietnam

Ali Sohani, *K.N. Toosi University of Technology*, Iran

Pankaj Kumar, *GBPUA&T*, India

Zehra Yigit, *Istanbul Technical University*, Turkey

Chunxiao Jiang, *Tsinghua University*, China

Omar Hiari, *German Jordanian University*, Jordan

Sangchoon Kim, *Dong-A University*, Korea

Mohammed El-Hajjar, *University of Southampton*, UK

Fahad Siddiqui, *Queen's University Belfast*, UK

Tiantai Deng, *Queen's University Belfast*, UK

Umar Ibrahim Minhas, *Queen's University Belfast*, UK

Donald Bailey, *Massey University*, New Zealand

S. B. Kotsiantis, *University of Patras*, Greece

Mohammed Mahfouz Elmogy, *Mansoura University*, Egypt

Guoru Ding, *PLA University of Science and Technology*, China

G. Rodrıguez Sanchez, *Universidad de Salamanca*, Spain

Hale Kirer Silva Lecuna, *Bandırma Onyedi Eylül University*, Turkey

Liping Feng, *Chongqing University*, China

A. Queiruga Dios, *Universidad de Salamanca*, Spain

Muhammad R. Ramli, *Kumoh National Institute of Technology*, Korea

M. Ezhilarasi, *Kumaraguru College of Technology*, India

Arda Gumusalan, *George Mason University*, USA

Samy S. Abu-Naser, *Al-Azhar University*, Palestine

William Sandham, *University of Strathclyde*, Scotland

Kezhi Li, *Imperial College London*, UK

Alberto Ochoa, *Juarez City University*, Mexico

Seyed Reza Kamel, *Islamic Azad University*, Iran

Olga B. Mora, *University of Guadalajara*, Mexico

Annette Payne, *Brunel University*, UK

Leila Shahmoradi, *University of Medical Sciences*, Iran

Omid Hamidie, *Hamedan University of Technology*, Iran

Leili Tapaka, *Hamedan University of Technology*, Iran

Cunlu Xu, *Lanzhou University*, China

Sanchita Basak, *Vanderbilt University*, USA

Keke Gai, *Beijing Institute of Technology*, China

Ali Jaoua, *Qatar University*, Qatar

Asad Waqar Malik, *NUST*, Pakistan

Pradeeban Kathiravelu, *ULisboa*, Portugal

Konstantinos M. Giannoutakis, *Information Technologies Institute*, Greece

Manoel C. Silva, *Federal University of Tocantins*, Brazil

Junhai Luo, *UESTC*, China

Nasir Saeed, *KAUST*, KSA

Haythem Bany Salameh, *Yarmouk University*, Jordan

Vishnupriya Kuppusamy, *University of Bremen*, Germany

Arnab Kumar Maity, *Texas A&M University*, USA

Sulma Rashid, *COMSATS University*, Pakistan

Syed Tahir Ali Jan, *Riphah International University*, Pakistan

Salima Harrat, *Ecole Normale Supérieure de Bouzaréah*, Algeria

Eslam Kamal Abdelreheem, *Microsoft*, USA

Ahmed Guessoum, *University of Science and Technology Houari Boumediene*, Algeria

Mahmoud Adnen, *Monastir Faculty of Science*, Tunisia

Damien Nouvel, *Inalco ERTIM*, France

Torsten Zesch, *University of DuisburgEssen*, Germany

Ke Zhang, *North China Electric Power University*, China

Shixing Chen, *Wayne State University*, USA

Shuzhe Wu, *Institute of Computing Technology*, China

Shan Xue, *Macquarie University*, Australia

Alireza Siadatan, *University of Toronto*, Canada

Abdolmajid Mousavi, *University of Lorestan*, Iran

Muhammad Nadzir Marsono, *UTM*, Malaysia

Nadeen Gebara, *Imperial College London*, UK

Mario P. Vestias, *ISEL*, Portugal

Khyamling Parane, *NITK*, India

Ramin Bashizade, *Sharif University of Technology*, Iran

Morteza Shahedifar, *Tabriz University*, Iran

Zhan Zhang, *University of Electronic Science and Technology of China*, China

Emilio Sirignano, *University of Modena and Reggio Emilia*, Italy

Jean-Fu Kiang, *National Taiwan University*, Taiwan

Yuntao Wu, *Wuhan Institute of Technology*, China

S. Shirvani Moghaddam, *Shahid Rajaei University*, Iran

Mohamed H. Gad-Elrab, *Max-Planck Institute for Informatics*, Germany

Hany H. Ammar, *West Virginia University*, USA

Gheith Abandah, *University of Jordan*, Jordan

Ahmed Lawgali, *Benghazi University*, Libya

Rania Maalej, *National School of Engineers of Sfax*, Tunisia

Mohamed E. Hussein, *Information Sciences Institute, USC*, USA

Mahmoud Khalifa, *University of Science and Technology*, China

Gridaphat Sriharee, *King Mongkut's University of Technology North Bangkok*, Thailand

Hang Dong, *University of Liverpool*, UK

Ali Mansour, *University of Bedfordshire*, UK

Fuad Jamour, *University of California*, USA

Majdi Rawashdeh, *PSUT*, Jordan

Lorenza Saitta, *Università del Piemonte Orientale*, Italy

Daniel Castro Silva, *University of Porto*, Portugal

Ahmad Al-Khalil, *University of Duhok*, Iraq

C. Xu, *University of Southampton*, UK

Montadar Abas Taher, *University of Diyala*, Iraq

Stefan Pratschner, *Vienna University of Technology*, Austria

Yunus Celik, *University of Sheffield*, UK

Moloud Abdar, *Deakin University*, Australia

Issam El Naqa, *WUSTL*, USA

Nicandro Cruz-Ramırez, *Universidad Veracruzana*, Mexico

Mariam Zomorodi-Moghadam, *Ferdowsi University of Mashhad*, Iran

Nebojsa Bacanin, *Singidunum University*, Serbia

Morshed U. Chowdhury, *Deakin University*, Australia

Abdellah Ezzati, *Hassan 1st University*, Morocco

Salem Omar Sati, *Misurata University*, Libya

Sakeena Javaid, *COMSATS University*, Pakistan

Katrina Issa Sundus, *University of Jordan*, Jordan

Rula Alrawashdeh, *Mutah University*, Jordan

Gilbert M. Gilbert, *University of Dar es Salaam*, Tanzania

Shengli Fu, *University of North Texas*, USA

Junshan Luo, *National University of Defense Technology*, China

N. S. Murthy, *ICMR, Ramaiah Medical College*, India

Yuteng Wu, *Illinois Institute of Technology*, USA

Karim Bouzoubaa, *Mohammed V University*, Morocco

Driss Namly, *Mohammed V University*, Morocco

Mohamed Amine Jerbi, *Higher Institute of Management of Tunis*, Tunisia

Hadhemi Achour, *Tunis University*, Tunisia

Mohamed Amine Cheragui, *Ahmed Draia University*, Algeria

Hussein Hussein, *Free University of Berlin*, Germany

Kathrein Abu Kwaik, *Gothenburg University*, Sweden

Mehrdad Mirzaei, *University at Albany*, USA

Juho Leinonen, *University of Helsinki*, Finland

Maria Joo Varanda, *Polytechnic Institute of Bragança*, Portugal

Duaa Alawad, *University of New Orleans*, USA

Lei Xu, *Nanjing University*, China

Anna Corazza, *Università di Napoli "Federico II"*, Italy

Luca Pascarella, *Delft University of Technology*, Netherlands

Lili Bo, *China University of Mining and Technology*, China

Izzat Alsmadi, *Texas A&M University*, USA

Abdelali Zakrani, *Hassan II University*, Morocco

JJCIT Annual List of Reviewers (2020)

Name, *Affiliation*, Country

Manish Kumar, *IIIT-Allahabad*, India
Josef J. Langerman, *University of Johannesburg*, South Africa
Leonardo Leite, *University of São Paulo*, Brazil
Jim Buchan, *Auckland University of Technology*, New Zealand
Nicolas Guelf, *University of Luxembourg*, Luxembourg
Rahul Yadav, *Harbin Institute of Technology*, China
Jakub Krzywda, *Ume University*, Sweden
Erica T. G. Sousa, *Federal Rural University of Pernambuco*, Brazil
Taskeen Zaidi, *Babasaheb Bhimrao Ambedkar University*, India
Nguyen Quang-Hung, *HCMUT University*, Vietnam
Mhamed Mataoui, *Ecole Militaire Polytechnique*, Algeria
Salima Mdhaffar, *Le Mans University*, France
Erik Cambria, *Nottingham Trent University*, UK
Mounir Ben Ayed, *University of Sfax*, Tunisia
Evangelia Kyrimi, *Queen Mary University of London*, UK
Amir Mohammad Shahsavarani, *Institute of PsychoBioSocioEconomic Sciences*, Iran
Alberto Cano, *Virginia Commonwealth University*, USA
Thang Mai, *Duy Tan University*, Vietnam
Jerry Chun-Wei Lin, *Western Norway University of Applied Sciences*, Norway
Asma Belhadi, *University of Science and Technology Houari Boumediene*, Algeria
V. M. Thakare, *Amravati University*, India
Cheng-Yuan Tang, *Huafan University*, Taiwan
Alaei Alireza, *Griffith University*, Australia
Mohamed Biniz, *University Sultan Moulay Sliman*, Morocco
Rania M. Ghoniem, *Prince Nourah University*, KSA
Pit Pichappan, *Digital Information Research Foundation*, India
Rasha Mamoun Elhassan, *King Khalid University*, KSA
Samir Boukil, *Chouaib Doukkali University*, Morocco
Ali Abdularahman Alani, *University of Diyala*, Iraq
Raed Abu Zitar, *Ajamn University*, UAE
Samaa M. Shohieb, *Mansoura University*, Egypt
Jianhua Liu, *BUCEA*, China
Monji Kherallah, *University of Sfax*, Tunisia
Manar Reyad Maraqa, *Al Balqa Applied University*, Jordan
Irfan Ullah, *University of Peshawar*, Pakistan
Shah Khusro, *University of Peshawar*, Pakistan
Hector Migalln, *Miguel Hernndez University*, Spain
Marcos L. Chaim, *University of São Paulo*, Brazil
Shahid Hussain, *COMSATS University*, Pakistan

Jinzhuo Liu, *Yunnan University*, China
Mahdi Moradian, *Missouri University of Science and Technology*, USA
Sangkil Kim, *Pusan National University*, South Korea
Hadi Hijazi, *ENSTA Bretagne*, France
Derar Hawatmeh, *JUST*, Jordan
Ali Reza Hazeri, *Islamic Azad University*, Iran
Panagiotis K. Gkonis, *National and Kapodistrian University of Athens*, Greece
Imtiaz Parvez, *Florida International University*, USA
Wahab Khawaja, *North Carolina State University*, USA
Dimitra I. Kaklamani, *National Technical University of Athens*, Greece
Mohammed A. A., *Wuhan University*, China
Lina Yao, *University of New South Wales*, Australia
Zhengjie Wang, *Shandong University of Science and Technology*, China
Chaoyang Jiang, *University of South Dakota*, USA
Jollen Chen, *Flowchain Foundation*, Singapore
Yunlong Lu, *Institute of Network Technology, Beijing University*, China
Kun Meng, *Beijing Information Science & Technology University*, China
Ping Yu, *University of Technology Sydney*, Australia
Zibin Zheng, *Sun Yat-sen University*, China
Marc Pilkington, *University of Burgundy*, France
Abdel Belaid, *LORIA*, France
Jabril Ramdan, *Aljafara University*, Libya
Muhammad Imran Razzak, *Deakin University*, Australia
Roslina Ibrahim, *UTM*, Malaysia
Nur Azaliah Abu Bakar, *UTM*, Malaysia
Dulani Meedeniya, *University of Moratuwa*, Sri Lanka
A. Spiteri Staines, University of Malta, Malta
Mohamed Bettaz, *Philadelphia University*, Jordan
Sepher Tabrizchi, *Institute for Research in Fundamental Sciences*, Iran
Ali Shahhoseini, *Qazvin Islamic Azad University*, Iran
Daniel Etiemble, *LRI*, France
Junaid Shuja, *COMSATS University*, Pakistan
Huangke Chen, *National University of Defense Technology*, China
Weidong Bao, *National University of Defense Technology*, China
Abdelkader Tami, *Sohar University*, Oman
Owoicho E. Ijiga, *University of the Witwatersrand*, South Africa
Felipe A. P., *Ghent University*, Belgium
Maryam K. Abboud, *Nahrain University*, Iraq
Mokhtar Keche, *LSI, USTO-MB*, Algeria
Kheira-Zineb Bousmaha, *Oran University 1 Ahmed Ben Bella*, Algeria
Ouafae Nahli, *Institute for Computational Linguistics*, Italy

Fernando Pinciroli, *Champagnat University*, Argentina
Tomas Cerny, *Baylor University*, USA
Fernando Asteasuain, *Universidad Nacional de Avellaneda*, Argentina
Jun Huang, *University of Chinese Academy of Sciences*, China
Suping Xua, *Nanjing University*, China
Jose M. Moyano, *University of Cordoba*, Spain
Jing Li, *University of New Haven*, USA
Jennifer O. Contreras, *Future University*, Philippines
Shadi Hilles, *Istanbul Okan University*, Turkey
Guoxi Liang, *Wenzhou Polytechnic*, China
Leila Ouahrani, *Blida University*, Algeria
Aluizio Haendchen Filho, *UNIVALI*, Brazil
John Nerbonne, *University of Freiburg*, Germany
Yassine Ben Ayed, *University of Sfax*, Tunisia
Abdul Halim Embong, *International Islamic University Malaysia*, Malaysia
Bassel Soudan, *University of Sharjah*, UAE
Akram M Zeki, *International Islamic University Malaysia*, Malaysia
Nik Nur Wahidah, *International Islamic University Malaysia*, Malaysia
Min Ma, *Google*, USA
Mohammad Ayoub Khan, *University of Bisha*, KSA
George Hadjichristofi, *Frederick University*, Cyprus
Karim Lounis, *Queen's University*, Canada
Mohammad Alshayeb, *KFUPM*, KSA
Nor Laily Hashim, *Universiti Utara Malaysia*, Malaysia
Martina Radilova, *University of Žilina*, Slovakia
Shiqing Xin, *Shandong University*, China
Fucang Jia, *Shenzhen Institutes of Advanced Technology*, China
Dan Liu, *East China University of Technology*, China
Patrik Kamencay, *University of Žilina*, Slovakia
Roman Jarina, *University of Žilina*, Slovakia

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

OCLC WorldCat

crossref Text and Data Mining

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS

Google scholar

ACADEMIA

BASE

فريق دعم هيئة التحرير

المحرر اللغوي

حيدر المومني

ادخال البيانات وسكرتير هيئة التحرير

إياد الكوز

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
Since 1991 تأسست
PSUT
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

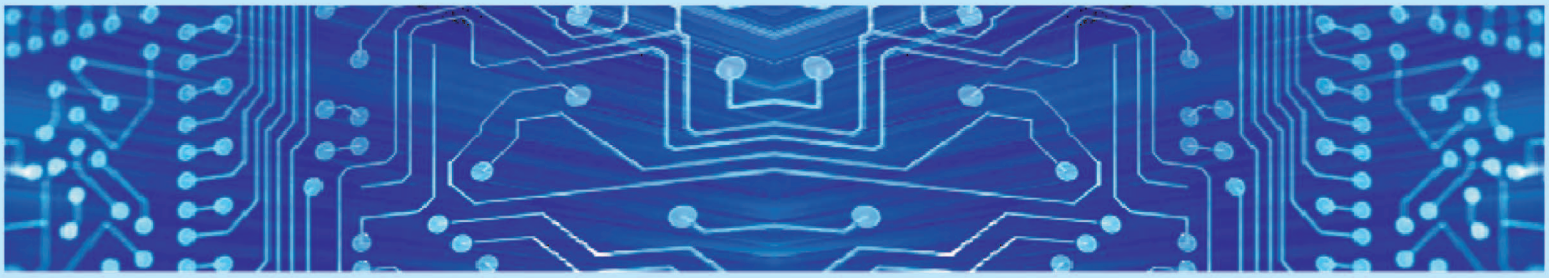
المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)
العدد ٤ المجلد ٦ كانون الأول ٢٠٢٠

JJCIT

الصفحات عنوان البحث

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



مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي