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

صندوق دعم البحث العلمي
Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology

September 2020 VOLUME 06 NUMBER 03
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

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

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

Jordanian Journal of Computers and Information Technology (JJCIT)

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

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

JJCIT Address

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

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

FULL-RATE SPACE-TIME BLOCK CODE FOR FOUR TRANSMIT ANTENNAS WITH LINEAR DECODING

Younes Djemamar[1], Saida Ibnyaich[1] and Abdelouhab Zeroual[1]

(Received: 3-Feb.-2020, Revised: 22-Apr.-2020, Accepted: 7-May-2020)

ABSTRACT

This paper proposes a new full-rate space-time block code for MIMO systems at four transmit antennas, which aims at minimizing the bit error rate with low-complexity decoding using ZF and MMSE linear detection techniques in Rayleigh fading channels. The purpose of this code is to optimize the space-time resources offered by MIMO systems and to ensure optimal exploitation of spectral resources. The idea is to take advantage of the direct sequence spread spectrum (DSSS) technique using the orthogonal codes of Walsh-Hadamard in order to ensure the orthogonality between all the symbol vectors to be transmitted and to construct an orthogonal and full-rate STBC code. BER performance versus Eb/No of the proposed STBC code is evaluated for MIMO 4×2, 4×4 and MISO 4×1 configurations in single-user detection mode, using 16QAM modulation, on Rayleigh's MIMO channels assumed to be quasi-static and not frequency selective, compared with those of the orthogonal STBC of Tarokh of 1/2 rate and the full-rate quasi-orthogonal STBC of Jafarkhani. The results of the simulations show that the proposed STBC code significantly improves BER performance while allowing a higher transmission rate with spectral efficiency of 4 bits/s/Hz and simple linear decoding using ZF or MMSE techniques.

KEYWORDS

MIMO system, STBC, Full-rate, BER, Zero forcing, MMSE.

1. INTRODUCTION

Space-time block coding is a very popular diversity approach, used to combat channel fading and minimize bit error rates [1]-[2]. Thanks to this diversity approach, a maximum diversity of $N_T \times N_R$ equal to the number of independent paths available can be obtained. Orthogonal space-time block codes (OSTBC) have been designed to obtain the maximum diversity order for a given number of transmit and receive antennas using a simple linear decoding algorithm at reception [3]. This has made these codes the most dominant forms of space-time codes, adopted in several wireless communication standards [4].

For an optimal exploitation of spectral resources, it is interesting to have an STBC code with unit rate. The code rate R_{STBC} is defined as the ratio between the number of transmitted symbols K per code word and the number of symbol durations T during which these symbols are transmitted $R = K/T$. The best STBC codes in the literature are those with a unit rate and achieve maximum diversity for a given number of transmit antennas. Alamouti's complex STBC code using two transmit antennas, proposed by S. Alamouti in 1998, is the only full-rate orthogonal complex code [5]. This code allows maximum diversity using a single antenna for reception. Alamouti code then is the only STBC code making it possible to achieve these two characteristics characterizing the optimal STBC code, with acceptable BER performance at reception [5]. Therefore, Alamouti code is a very special code and one of the most widely adopted codes.

Increasing the number of both transmit and receive antennas improves system performance and reduces bit error rate [6]-[7]. In this setting, in 1999, V. Tarokh *et al.* had the initial idea to generalize the concept of Alamouti to a number of transmit antennas more than two by reducing the code rate ($R_{STBC} < 1$) and by leaving diversity and orthogonality unchanged [8]. Consequently, these lower rate codes alter the transmission rate and degrade the spectral efficiency of the MIMO system [9]. Full-rate quasi-orthogonal codes (QOSTBCs) have been proposed to achieve total transmission rate for wireless systems with more than two transmit antennas, at the cost of losing some of the diversity and increasing the decoding complexity. However, these codes affect performance in error rates at low and

1. Y. Djemamar*, S. Ibnyaich and A. Zeroual are with Instrumentation, Signals and Physical Systems (I2SP), Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco. E-mail: *younes.djemamar@gmail.ma

high SNR, which limits their use [10]. Many unit rate codes have been developed, including codes based on Alamouti code with switching between antenna groups, linear dispersion codes and DAST diagonal algebraic codes, allowing maximum diversity. However, some of them present higher complexity in coding and/or decoding or even limited BER performance. Full-rate linear dispersion algebraic codes have been the subject of several studies until today; however, the disadvantage of this type of STBC codes lies in the difficulty of optimizing the dispersion matrices (they are dependent on the constellation) in order to maximize diversity, as well as high complexity in coding. This paper proposes a new full-rate space-time block code for MIMO systems for four transmit antennas, which aims at minimizing the bit error rate with low-complexity decoding using ZF linear detection technique in Rayleigh fading channels. The purpose of this code is to optimize the space-time resources offered by MIMO systems and therefore allow optimal exploitation of spectral resources. The paper is organized as follows: in Section 2, the system model of STBC-MIMO is described. Next, main existing STBC codes for MIMO systems of four transmit antennas are presented in Section 3. In Section 4, the proposed STBC code for four transmit antennas is presented. In Section 5, the simulation methodology is discussed; then, results and analysis are presented. Finally, Section 6 concludes this paper.

2. SYSTEM MODEL

The idea behind MIMO systems based on space-time block codes (STBC) is to transmit data in such a way as to guarantee great diversity, while allowing a simple decoding process. The STBC code consists of blocks. Each block is coded according to precise rules and independently of the other blocks. The idea is to send each block of symbols to the N_T transmit antennas. A decoding error occurring in one block does not jeopardize other blocks.

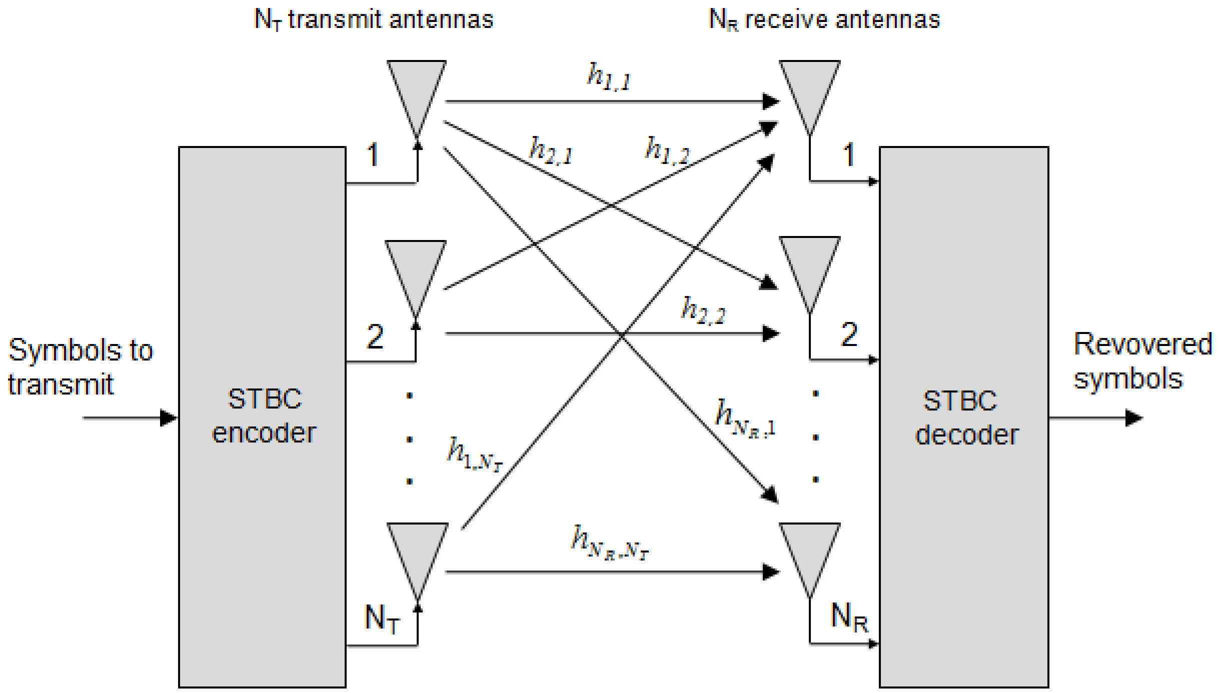


Figure 1. STBC-MIMO system.

A space-time block code is generally represented by an $N_T \times T$ matrix, where each line represents a transmit antenna and each column represents a block of symbols of block duration:

$$C = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1T} \\ s_{21} & s_{22} & \cdots & s_{2T} \\ \vdots & \vdots & \vdots & \vdots \\ s_{N_T 1} & s_{N_T 2} & \cdots & s_{N_T T} \end{pmatrix} \quad (1)$$

where s_{ij} is the coded symbol on the i^th^ transmit antenna at the j^th^ time slot.

For a MIMO system with N_T transmit antennas and N_R receive antennas, the received signal y_j at each moment on the j[th] receive antenna is the sum of the symbols derived from the N_T transmitted signals:

$$y_i = \sum_{i=1}^{N_T} h_{i,j} s_j + n_i \tag{2}$$

where $h_{i,j}$ is the attenuation and phase shift (transfer function) of the non-selective frequency channel between the j[th] transmit antenna and i[th] receive antenna and n_i is the additive noise. The complex matrix H of the channel can then be written as follows:

$$H = \begin{pmatrix} h_{1,1} & . & .. & h_{1,N_T} \\ . & h_{2,2} & .. & . \\ \vdots & \vdots & \vdots & \vdots \\ h_{N_R,1} & h_{N_R,2} & .. & h_{N_R,N_T} \end{pmatrix} \tag{3}$$

The MIMO signal model can be described as:

$$y = H.C + n \tag{4}$$

where y and n are respectively receive and noise vectors of size $N_R \times 1$, H is the channel matrix of size $N_R \times N_T$ and c is the transmitted vector of size $N_T \times 1$.

Each STBC code is characterized by a code rate R_{STBC}. The code rate R_{STBC} is defined as the ratio between the number of transmitted symbols K per code word and the number of symbol durations T during which these symbols are transmitted:

$$R = K / T \tag{5}$$

The STBC C code is orthogonal if it satisfies the following criteria:

$$C.C^H = A.I_n \tag{6}$$

where C^H denotes the Hermitian transpose of the code matrix, A is a real coefficient and I_n is the identity matrix of size n×n (n ϵ IN).

3. Main Existing STBC For Four Transmit Antennas

3.1 Orthogonal-STBC Codes of Tarokh

Tarokh *et al.* proposed complex orthogonal space-time block codes in [11] for four transmit antennas. These codes provide maximum diversity; however, they have the disadvantage of having a rate lower than one. The matrices of codes C_4 and X_4 of rates R_{STBC} = 1/2 and R_{STBC} = ¾, respectively, for four transmit antennas, are as follows:

$$C_4 = \begin{pmatrix} s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^* \\ s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^* \\ s_3 & -s_4 & s_1 & s_2 & s_3^* & -s_4^* & s_1^* & s_2^* \\ s_4 & s_3 & -s_2 & s_1 & s_4^* & s_3^* & -s_2^* & s_1^* \end{pmatrix} \tag{7}$$

$$X_4 = \begin{pmatrix} s_1 & -s_2^* & \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} \\ s_2 & s_1^* & \frac{s_3^*}{\sqrt{2}} & -\frac{s_3^*}{\sqrt{2}} \\ \frac{s_3}{\sqrt{2}} & \frac{s_3}{\sqrt{2}} & \frac{-s_1 - s_1^* + s_2 - s_2^*}{2} & \frac{s_2 + s_2^* + s_1 - s_1^*}{2} \\ \frac{s_3}{\sqrt{2}} & -\frac{s_3}{\sqrt{2}} & \frac{-s_2 - s_2^* + s_1 - s_1^*}{2} & -\frac{s_1 + s_1^* + s_2 - s_2^*}{2} \end{pmatrix} \tag{8}$$

where s_1, s_2, s_3 and s_4 are the symbols to be transmitted and $(.)^*$ denotes the complex conjugate.

The channel matrices obtained are therefore orthogonal. Consequently, the decoding of the symbols can be done simply with the maximum likelihood technique while reducing the processing complexity.

3.2 Quasi-OSTBC Codes

Interesting compromises for MIMO systems at four transmit antennas were proposed by H. Jafarkhani in [12] and Tirkkonen in [13]. The idea is to create quasi-orthogonal complex codes admitting a simplified maximum likelihood decoding (but more complex than the decoding of an orthogonal code).

The matrices of full-rate quasi-orthogonal space-time block codes C_{Jaf} and C_{Tir} proposed by Jafarkhani and Tirkkonen, respectively, are as follows:

$$C_{Jaf} = \begin{pmatrix} s_1 & -s_2^* & -s_3^* & s_4 \\ s_2 & s_1^* & -s_4^* & -s_3 \\ s_3 & -s_4^* & s_1^* & -s_2 \\ s_4 & s_3^* & s_2^* & s_1 \end{pmatrix} \tag{9}$$

$$C_{Tir} = \begin{pmatrix} s_1 & -s_2^* & s_3 & -s_4^* \\ s_2 & s_1^* & s_4 & s_3^* \\ s_3 & -s_4^* & s_1 & -s_2^* \\ s_4 & s_3^* & s_2 & s_1^* \end{pmatrix} \tag{10}$$

Both full-rate quasi-orthogonal codes of Jafarkhani and Tirkkonen have the same BER performance at low and high SNR in Rayleigh fading channels [14].

The orthogonal STBC developed by Tarokh with a rate of 1/2 and the full-rate quasi-orthogonal STBC of Jafarkhani, for four transmit antennas, are widely used and most adopted in modern and futuristic digital wireless communication systems, due to their lower complexities in coding and decoding. For this reason, they constitute the reference codes for simulating the performance of our proposed STBC.

4. PROPOSED STBC CODE

The principle of proposed STBC coding in this research aims at exploiting the technique of spectrum spreading by direct sequences (DSSS) using the orthogonal codes of Walsh-Hadamard, in order to ensure orthogonality between all the symbol vectors to be transmitted and build an orthogonal STBC code at a rate of 1.

The idea is that each of the four information symbols to be transmitted is multiplied in the time domain by its own Walsh-Hadamard code (correlation product). The four resulting symbols to be sent are applied to a square matrix of order 4 for real space-time coding, where each of these symbols is different from all adjacent symbols and is orthogonal with them, as shown in Table 1. Our design criteria for the proposed code do not depend on the used constellation.

Walsh-Hadamard codes are not very long or PN type for a significant spread spectrum. Walsh-Hadamard codes are adopted in our designed STBC-MIMO system for their orthogonality, ease of generation and simplicity of implementation. They provide optimal performance in the presence of synchronous transmission.

Figure 2 presents the general scheme of the proposed transmit system. It comprises four main subsystems: modulator, serial-parallel converter, Walsh-Hadamard coder and space-time coder in blocks. After mapping operation, data to be transmitted is converted from series into parallel from with four symbols at the output. Then, these four symbols are correlated (multiplication in the time domain) with their different four own Walsh-Hadamard codes of the same length, as shown in Figure 3. We can express this in mathematical form by writing the Walsh-Hadamard spreading code vectors and the information symbol vectors successively in the matrices C and s:

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \end{bmatrix} \tag{11}$$

$$s = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} \tag{12}$$

To carry out the matrix multiplication of the code vectors of C with the matrix $\boldsymbol{s}$ of the information symbol vectors, it will be advantageous to put the matrix C in diagonal form:

$$C = diag(c_1 \quad c_2 \quad c_3 \quad c_4) \tag{13}$$

The implicit form of multiplication can then be represented by:

$$S = s.C = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \end{pmatrix} \tag{14}$$

Finally, the reconverted symbols from parallel form into serial form apply to the space-time block coder for a real space-time coding in accordance with the following square coding matrix 4×4, where each of these symbols is different from all adjacent symbols.

Table 1. Space-time coding of the proposed 4×4 code.

	Antenna 1	**Antenna 2**	**Antenna 3**	**Antenna 4**
Time t	s_1	s_2	s_3	s_4
Time t+T	s_2	s_1	s_4	s_3
Time t+2T	s_3	s_4	s_1	s_2
Time t+3T	s_4	s_3	s_2	s_1

The number of symbols transmitted K per codeword is equal to the number of time slot T necessary to transmit these symbols; $K = T = 4$, then, $R_{STBC} = K/T = 1$. The proposed STBC code for four transmit antennas is therefore full-rate and is presented as follows:

$$C_{STBC,4} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2 & s_1 & s_4 & s_3 \\ s_3 & s_4 & s_1 & s_2 \\ s_4 & s_3 & s_2 & s_1 \end{pmatrix} \tag{15}$$

with $s = x_i.c_j$, where x_i is the information symbol to be transmitted and c_j is the Walsh-Hadamard spreading code specific to the information symbol x_i. The matrix of the proposed STBC code is symmetrical:

$$C_{STBC} = C_{STBC}^{H} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2 & s_1 & s_4 & s_3 \\ s_3 & s_4 & s_1 & s_2 \\ s_4 & s_3 & s_2 & s_1 \end{pmatrix} \tag{16}$$

Each symbol s of the proposed STBC code is a correlation product of a useful symbol with its own Walsh-Hadamard code. The four signals obtained from the four correlation products are orthogonal to each other. The proposed STBC coding scheme is shown in Figure 3.

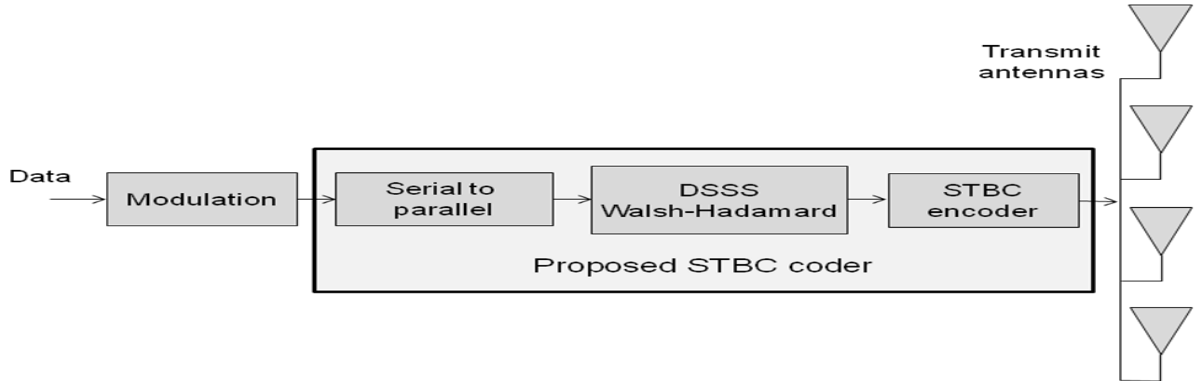


Figure 2. Block diagram of the proposed STBC-MIMO system in transmission.

As mentioned above, the design criteria for the proposed STBC code do not depend on the constellation used and the useful symbol x_i can be from a real or complex constellation.

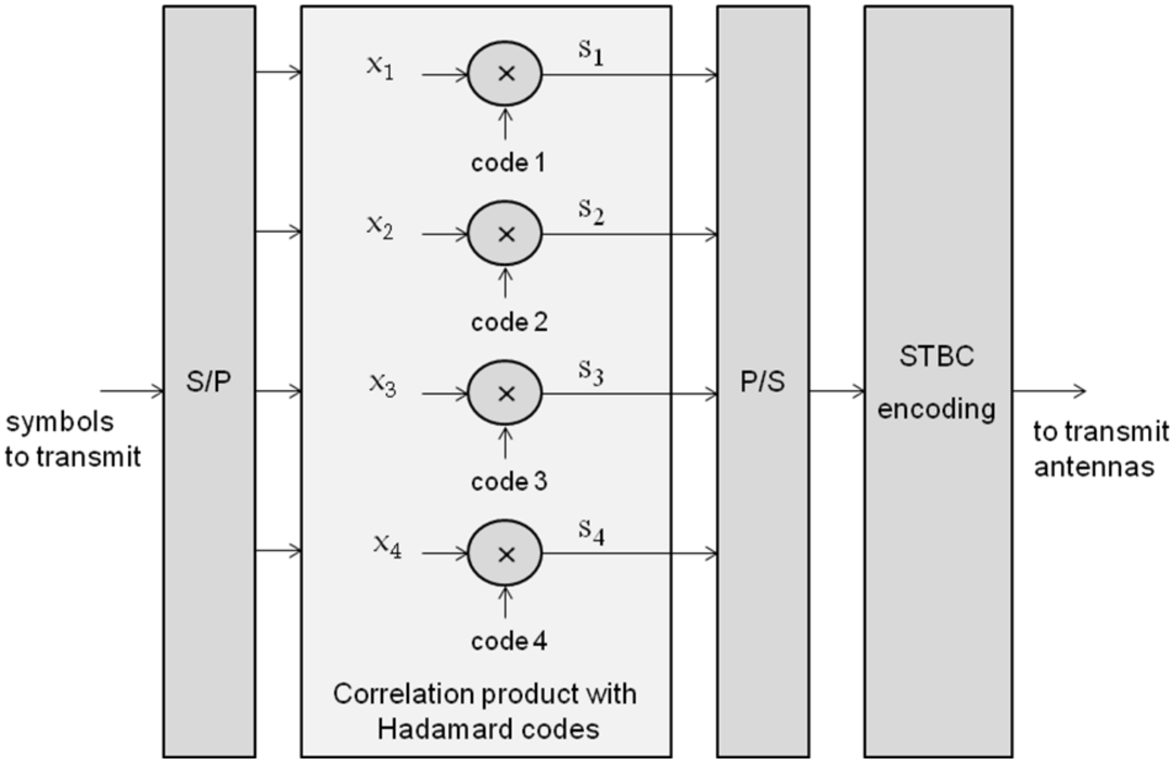


Figure 3. Principle of the proposed STBC coding.

Walsh-Hadamard codes of a length n are constructed from a Hadamard matrix of an order n. An example of the 8-bit Hadamard codes used in this study is shown in Figure 4.

The length of the 8-bit Walsh-Hadamard orthogonal codes was adopted in our proposed STBC code for simulations so that they ensure the orthogonality of the STBC code, and provide a better compromise between the used frequency band and the processing complexity especially at decoding. W-H sequences of length greater than 8 can be used (16 bit, 64 bit…).

$$H_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

code 1 code 2 . . . code 8

Figure 4. 8-bit Walsh-Hadamard codes from Hadamard matrix of order 8.

Due to this orthogonality property, cross-correlation between the four symbols s_1, s_2, s_3 and s_4 transmitted simultaneously every four time slots is zero due to perfect synchronization of their transmission.

$s_1(n)$, $s_2(n)$, $s_3(n)$ and $s_4(n)$ are orthogonal to each other over the interval $[n_1, n_8]$ and their scalar product is zero :

$$\sum_{n=n_1}^{n_8} s_1(n)s_2(n) = 0 \tag{17}$$

$$\sum_{n=n_1}^{n_8} s_1(n)s_3(n) = 0 \tag{18}$$

$$\sum_{n=n_1}^{n_8} s_1(n)s_4(n) = 0 \tag{19}$$

$$\sum_{n=n_1}^{n_8} s_2(n)s_3(n) = 0 \tag{20}$$

$$\sum_{n=n_1}^{n_8} s_2(n)s_4(n) = 0 \tag{21}$$

$$\sum_{n=n_1}^{n_8} s_3(n)s_4(n) = 0 \tag{22}$$

Similarly, all the columns of the code matrix are orthogonal to each other:

$$\sum_{n=n_1}^{n_8} \left(s_1(n)s_2(n)+s_2(n)s_1(n)+s_3(n)s_4(n)+s_4(n)s_3(n)\right)=0 \tag{23}$$

$$\sum_{n=n_1}^{n_8} \left(s_1(n)s_3(n)+s_2(n)s_4(n)+s_3(n)s_1(n)+s_4(n)s_2(n)\right)=0 \tag{24}$$

$$\sum_{n=n_1}^{n_8} \left(s_1(n)s_4(n)+s_2(n)s_3(n)+s_3(n)s_2(n)+s_4(n)s_1(n)\right)=0 \tag{25}$$

$$\sum_{n=n_1}^{n_8} \left(s_2(n)s_3(n)+s_1(n)s_4(n)+s_4(n)s_1(n)+s_3(n)s_2(n)\right)=0 \tag{26}$$

$$\sum_{n=n_1}^{n_8} \left(s_2(n)s_4(n)+s_1(n)s_3(n)+s_4(n)s_2(n)+s_3(n)s_1(n)\right)=0 \tag{27}$$

$$\sum_{n=n_1}^{n_8} \left(s_3(n)s_4(n)+s_4(n)s_3(n)+s_1(n)s_2(n)+s_2(n)s_1(n)\right)=0 \tag{28}$$

Consequently, the STBC 4×4 code achieves the following orthogonality criterion:

$$C_{STBC}.C_{STBC}^{H} = (|s_1|^2+|s_2|^2+|s_3|^2+|s_4|^2).\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix} \tag{29}$$

where C^H denotes the transposed matrix of the code. The code is then orthogonal.

The orthogonality property of the proposed STBC code allows spatial orthogonality between all the symbol vectors to be transmitted. The proposed orthogonal STBC code also provides inter-channel orthogonality between different transmit antennas. Similarly, the shape of the transmitted symbols (resulting from DSSS) allows a higher resistivity of the signal to interference in the channel and a more robust signal on reception.

The proposed system at the reception performs the inverse operation of the transmit system, as shown in Figure 5, by using one of the linear detection techniques ZF or MMSE.

The equivalent channel matrices obtained $H_{4\times1}$, $H_{4\times2}$ and $H_{4\times4}$ respectively of the proposed STBC code in MISO 4×1, MIMO 4×2 and MIMO 4×4 configurations, for four time slots, are represented below:

$$H_{4\times1} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 \\ h_2 & h_1 & h_4 & h_3 \\ h_3 & h_4 & h_1 & h_2 \\ h_4 & h_3 & h_2 & h_1 \end{pmatrix} \quad (30)$$

where h_j is the complex sub-channel coefficient between the j[th] transmit antenna and the receive antenna.

$$H_{4\times2} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ h_{1,2} & h_{1,1} & h_{1,4} & h_{1,3} \\ h_{2,2} & h_{2,1} & h_{2,4} & h_{2,3} \\ h_{1,3} & h_{1,4} & h_{1,1} & h_{1,2} \\ h_{2,3} & h_{2,4} & h_{2,1} & h_{2,2} \\ h_{1,4} & h_{1,3} & h_{1,2} & h_{1,1} \\ h_{2,4} & h_{2,3} & h_{2,2} & h_{2,1} \end{pmatrix} \quad (31)$$

$$H_{4\times4} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ h_{3,1} & h_{3,2} & h_{3,3} & h_{3,4} \\ h_{4,1} & h_{4,2} & h_{4,3} & h_{4,4} \\ h_{1,2} & h_{1,1} & h_{1,4} & h_{1,3} \\ h_{2,2} & h_{2,1} & h_{2,4} & h_{2,3} \\ h_{3,2} & h_{3,1} & h_{3,4} & h_{3,3} \\ h_{4,2} & h_{4,1} & h_{4,4} & h_{4,3} \\ h_{1,3} & h_{1,4} & h_{1,1} & h_{1,2} \\ h_{2,3} & h_{2,4} & h_{2,1} & h_{2,2} \\ h_{3,3} & h_{3,4} & h_{3,1} & h_{3,2} \\ h_{4,3} & h_{4,4} & h_{4,1} & h_{4,2} \\ h_{1,4} & h_{1,3} & h_{1,2} & h_{1,1} \\ h_{2,4} & h_{2,3} & h_{2,2} & h_{2,1} \\ h_{3,4} & h_{3,3} & h_{3,2} & h_{3,1} \\ h_{4,4} & h_{4,3} & h_{4,2} & h_{4,1} \end{pmatrix} \quad (32)$$

where $h_{i,j}$ is the complex coefficient of sub-channel between the j[th] transmit antenna and the i[th] receive antenna. Assuming that the elements $h_{i,j}$ of the matrix H are independent and identically distributed, and then the columns of the matrix H are independent, the detection of the transmitted symbols from the received vector can be carried out simply by using (ZF or MMSE) linear detection techniques. These two techniques ZF and MMSE consist in applying to the received vector y; respectively, the equalization matrices W_{ZF} and W_{MMSE} are as follows:

$$W_{ZF} = (H^H H)^{-1} H^H \tag{33}$$

$$W_{MMSE} = (H^H H + \sigma^2 I)^{-1} H^H \tag{34}$$

where $(.)^H$ denotes the Hermitian transpose operation and σ^2 is the statistical information of the noise. The estimate of the transmitted signal vector is then given by:

$$\hat{s}_{ZF} = W_{ZF} . y \tag{35}$$

$$\hat{s}_{MMSE} = W_{MMSE} . y \tag{36}$$

Thereafter, the information symbols can be recovered, after STBC decoding, by multiplying each by its own Walsh-Hadamard code (correlation product), as shown in Figure 6.

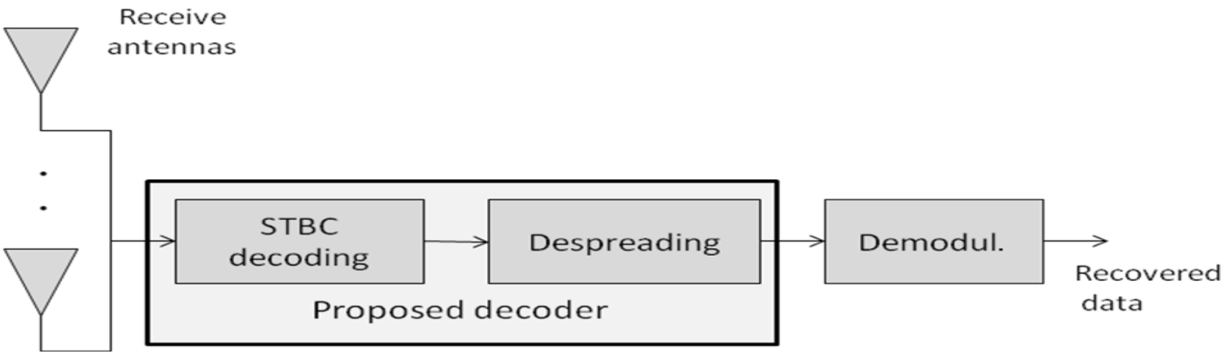


Figure 5. Proposed reception scheme.

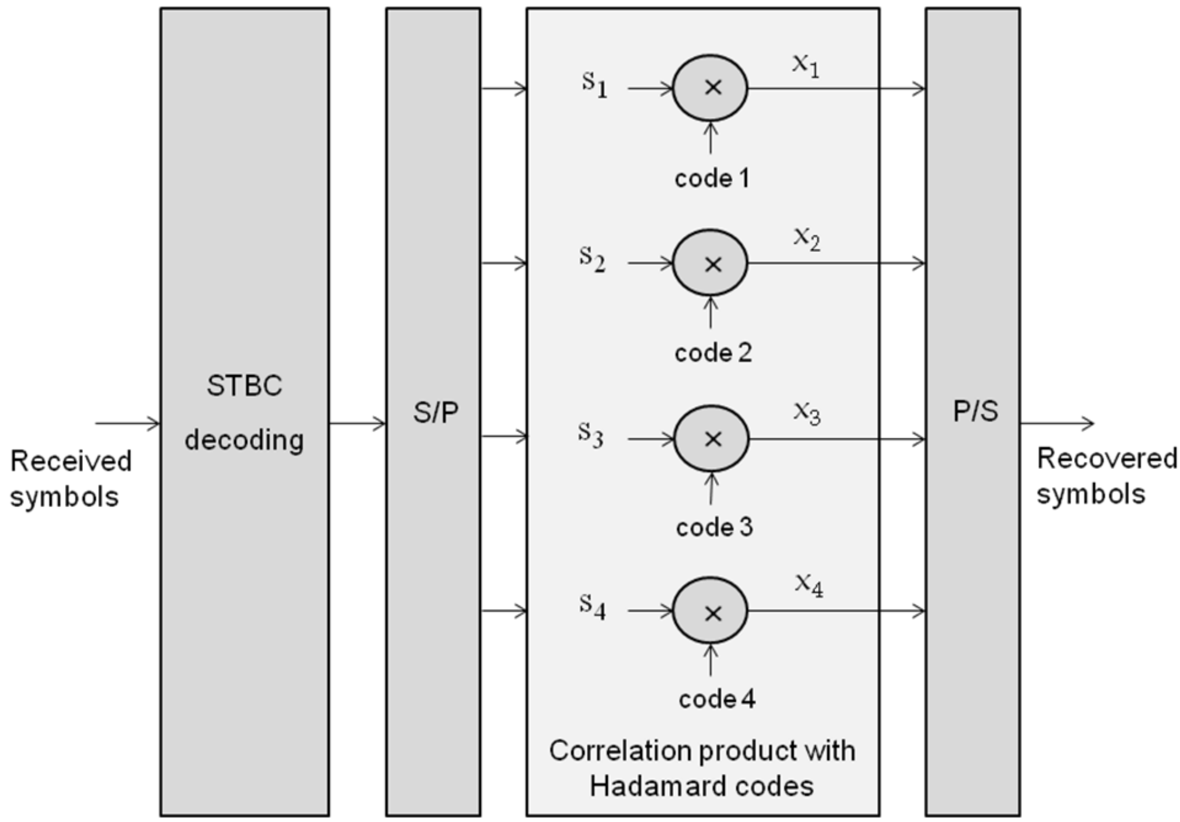


Figure 6. Principle of the proposed STBC decoding.

5. RESULTS AND DISCUSSION

The performance of the proposed code is evaluated in the bit error rate (BER) *versus* signal to noise ratio per bit (Eb/No) with 16QAM modulation in MATLAB using the quasi-static Rayleigh channel model. The MIMO channels are assumed to be frequency non-selective and not correlated.

The four Walsh-Hadamard codes used for each four symbols to transmit in the simulated proposed STBC code that are of 8-bit length are: [1 -1 -1 1 1 -1 -1 1], [1 -1 1 -1 -1 1 1 -1 1], [1 -1 1 -1 -1 1 -1 1] and [1 1 -1 -1 -1 -1 1 1], which are shown in Figure 4. Detection is in single-user mode and the channel is assumed perfectly estimated.

Figure 7 and Figure 8 show BER performances obtained *versus* the signal to noise ratio per bit (Eb/No) of the proposed STBC code for four transmit antennas with N_R = 1, 2 and 4 respectively using ZF and MMSE decoding, compared with the Tarokh orthogonal code C_4 of 1/2 rate with N_R = 1 and the unit rate quasi-orthogonal code C_{Jaf} with N_R = 1, using 16QAM modulation. It is clear from both figures that the proposed STBC code using ZF or MMSE decoding offers much better BER performance compared to the Tarokh orthogonal code C_4 of 1/2 rate and the full-rate quasi-orthogonal code of Jafarkhani C_{Jaf} using the same modulation 16QAM. Indeed, the proposed STBC code with N_R = 4 has better performance, then comes the proposed code with N_R = 2, afterwards the proposed code with N_R = 1, then the orthogonal code of Tarokh C_4 and finally the quasi-orthogonal code of Jafarkhani C_{Jaf}. We also note the significant difference between the BER curves of the proposed code in the MISO system having a theoretical diversity of 4 and the MIMO systems having the higher theoretical diversities of 8 and 16. The obtained results of the performances of the MIMO and MISO systems with the proposed STBC code confirm the theoretical performances in diversity and in decoding based on linear processing ZF and MMSE. In fact, the more the number of receive antennas increases, the lower the curve of the bit error rate becomes.

As shown in Figure 7, the proposed STBC code using ZF decoding with N_R = 1 presents a slight reduction in the error rate in very low signal-to-noise ratios (Eb/No≤ 5dB) and a noticeable reduction in the ratios 5db< Eb/No< 18dB; afterwards, its BER curve becomes above that of the OSTBC code C_4 until Eb/No = 25dB. For MIMO systems with N_R = 2 and 4, the proposed STBC code using ZF decoding presents a significant reduction in the error rate at low and high signal-to-noise ratios compared to the Tarokh orthogonal code and the quasi-orthogonal code of Jafarkhani, as shown in Figure 7. In the case of MMSE decoding, the proposed code with N_R = 1 presents a noticeable reduction in the error rate at low and high signal-to-noise ratios until Eb/No = 24dB compared to the Tarokh code and a significant reduction in the error rate at low and high signal-to-noise ratios compared to the quasi-orthogonal code of Jafarkhani, as shown in Figure 8. For MIMO systems with N_R = 2 and 4, the proposed STBC code using MMSE decoding presents a significant reduction in the error rate at low and high signal-to-noise ratios compared to the Tarokh orthogonal code and the quasi-orthogonal code of Jafarkhani, as shown in Figure 8.

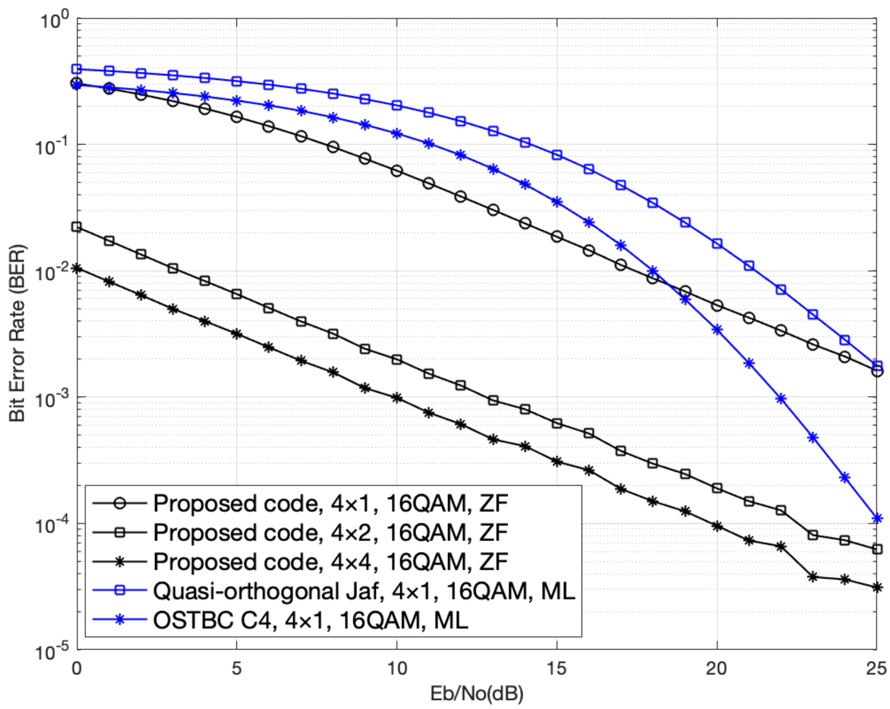


Figure 7. BER performance with 16QAM modulation for ZF decoding.

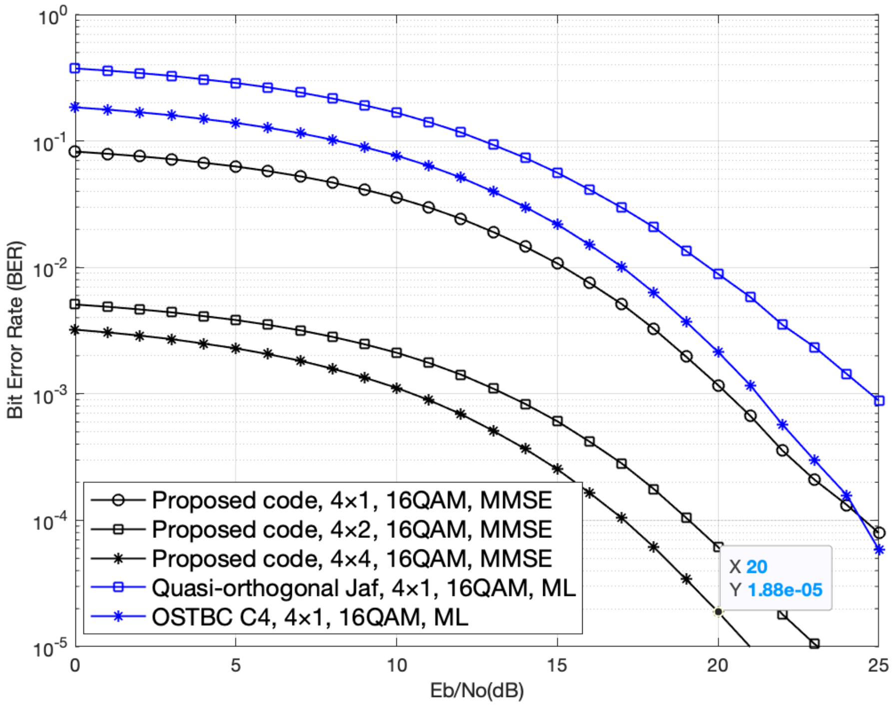


Figure 8. BER performance with 16 QAM modulation for MMSE decoding.

Figure 9 and Figure 10 show the BER performances of the STBC codes for a spectral efficiency η=4 bit/s/Hz using ZF and MMSE decoding, respectively. We note that the best performances are obtained by the systems with the proposed STBC code combining the 16QAM modulation. Indeed, the proposed STBC code with $N_R = 1$ having a theoretical diversity of 4 and using ZF decoding presents good performances in BER at low and high SNR compared to the quasi-orthogonal code of Jafarkhani and at low and high SNR up to Eb/No = 21dB compared to the Tarokh orthogonal code, as shown in Figure 9. In fact, the Tarokh orthogonal code C_4 of 1/2 rate with a theoretical diversity of 4 is associated with the 64QAM modulation which is less robust than the 16QAM modulation; hence, loss of performance (translation of its BER curve upwards) occurs. For MMSE decoding, the proposed STBC code in MISO and MIMO systems presents good performances in BER at low and high SNR compared to the Tarokh orthogonal code and the quasi-orthogonal code of Jafarkhani, as shown in Figure 10.

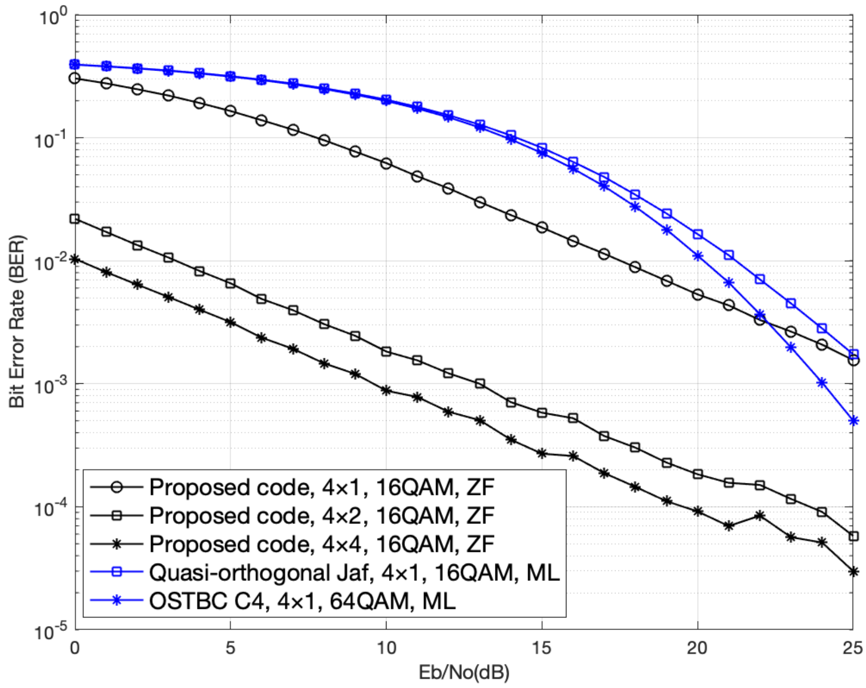


Figure 9. BER performance at η = 4 bit/s/Hz for ZF decoding.

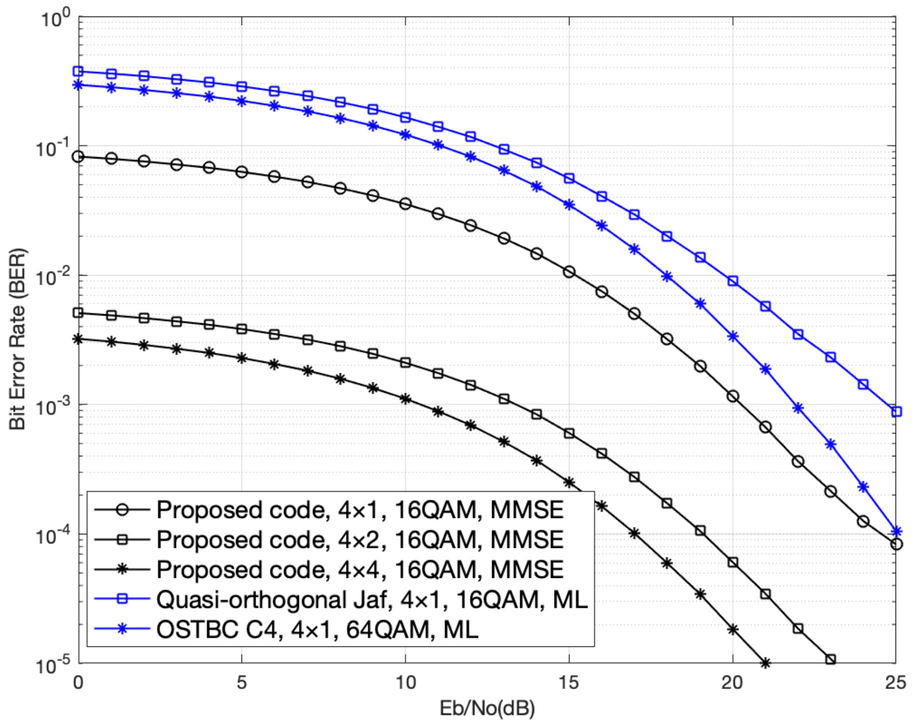


Figure 10. BER performance at η = 4 bit/s/Hz for MMSE decoding.

The proposed STBC code in MISO and MIMO systems in case of MMSE decoding provides much better BER performance than in the case of ZF.

6. Conclusions

This paper proposes a new full-rate space-time block code for MIMO systems with four transmit antennas, which aims at minimizing the bit error rate with low-complexity decoding using ZF and MMSE linear detection techniques in Rayleigh fading channels. The simulation results show that the proposed STBC code significantly improves BER performance while allowing simple linear decoding using ZF or MMSE techniques and higher transmission rate with a spectral efficiency of 4 bits/s/Hz, in a Rayleigh channel assumed to be quasi-static, not frequency selective and spatially uncorrelated. This proposed code allows an optimal exploitation of space-time resources offered by MIMO wireless systems based on space-time coding and therefore, an optimal exploitation of spectral resources.

References

[1] Y. A. Eldemerdash, O. A. Dobre and B. J. Liao, "Blind Identification of SM and Alamouti STBC-OFDM Signals," IEEE Transactions on Wireless Communications, vol. 14, no. 2, pp. 972-982, 2015.

[2] S. D. Santumon and B. R. Sujatha, "Space-time Block Coding (STBC) for Wireless Networks," International Journal of Distributed and Parallel Systems (IJDPS), vol. 3, no. 4, 2012.

[3] R. Biradar and G. Sadashivappa, "Study and Analysis of 2×2 MIMO Systems for Different Modulation Techniques using MATLAB," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 7, 2015.

[4] H. Sinha, M. R. Meshram and G. R. Sinha, "Performance Analysis of MIMO-STBC with Orthogonal Uncorrelated Channel," I-Manager's Journal on Wireless Communications Networks, vol. 6, no. 1, pp. 22-30, 2017.

[5] P. Pathak and R. Pandey, "A Novel Alamouti STBC Technique for MIMO System Using 16-QAM Modulation and Moving Average Filter," International Journal of Engineering Research and Applications, vol. 4, no. 8 (version 6), pp. 49-55, 2014.

[6] E. Ghayoula and R. Ghayoula, "Bit Error Rate Performance of MIMO Systems for Wireless Communications," World Academy of Science, Engineering and Technology, Electrical and Computer Engineering, vol. 2, no. 1, 2015.

[7] E. T. Tchao, K. Diawuo, W. K. Ofosu and E. Affum, "Analysis of MIMO Systems Used in Planning a 4G-WiMAX Network in Ghana," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 4, no. 7, 2013.

[8] Y. S. Cho, J. Kim, W. Y. Yang and C. Kang, MIMO-OFDM Wireless Communications with MATLAB, 1st Ed., Wiley-IEEE Press, 2010.

[9] P. Rabiei, N. Al-Dhahir and R. Calderbank, "New Rate-2 STBC Design for 2 TX with Reduced-Complexity Maximum Likelihood Decoding," IEEE Trans. on Wireless Communications, vol. 8, no. 2, pp. 1803–1813, 2009.

[10] F. Wang, A. Ghosh, C. Sankaran and S. Benes, "WiMAX System Performance with Multiple Transmit and Multiple Receive Antennas," Proc. IEEE 65th Vehicular Technology Conf. (VTC2007), pp. 2807-2811, 2007.

[11] V. Tarokh, H. Jafarkhani and R. A. Calderbank, "Space-time Block Codes from Orthogonal Designs," IEEE Transactions on Information Theory, vol. 45, pp. 1456–1467, July 1999.

[12] H. Jafarkhani, "A Quasi-orthogonal Space-time Block Code," IEEE Transactions on Communications, vol. 49, no. 1, pp. 1–4, January 2001.

[13] O. Tirkkonen, A. Boariu and A. Hottinen, "Minimal Non-orthogonality Rate-1 Space-time Block Code for 3+ Tx Antennas," Proceedings of IEEE International Symposium on Spread Spectrum Techniques and Applications, ISSTA 2000, pp. 429–432, Sept. 2000.

[14] V. L. Nir, Etude et Optimisation des Systèmes Multi-antennes Associés à des Modulations Multiporteuses, Mémoire de Doctorat (PhD Thesis), INSA de Rennes, 2004.

ملخص البحث:

تقترح هذه الورقة شيفرة مكانية-زمانية جديدة، ذات معدل كامل، مؤلّفة من وحدات، للأنظمة متعددة المداخل متعددة المخارج التي تستخدم أربعة هوائيات إرسال؛ بهدف التقليل ما أمكن من معدل الخطأ في البتّات (BER)، باستخدام إزالة تشفير منخفضة التعقيد عن طريق تقنيتي الكشف المعروفتين: (ZF) و(MMSE)، وهما من تقنيات الكشف الخطي في قنوات رايلي (Rayleigh) المضمحلة. ويتمثل الغرض من تلك الشيفرة في أمثلة الموارد المكانية-الزمانية التي توفرها الأنظمة متعددة المداخل متعددة المخارج، وضمان الاستغلال الأمثل للموارد الطيفية. وتكمن الفكرة في الاستفادة من تقنية الطيف المنتشر بالتتابع المباشر (DSSS) باستخدام شيفرات "والش- هادامارد" من أجل ضمان التعامد بين جميع متجهات الرموز المراد إرسالها، ومن ثم بناء شيفرة مكانية-زمانية، مؤلفة من وحدات، متعامدة و ذات معدل كامل. تم تقييم أداء الشيفرة المقترحة من حيث معدل الخطأ في البتّات (BER) لكلٍ من تشكيلات الأنظمة متعددة المداخل متعددة المخارج (2x4 و 4x4) والنظام متعدد المداخل وحيد المخرج (1x4) في نمط الكشف ذي المستخدم الواحد، باستخدام تعديل (16 QAM) على قنوات "رايلي" متعددة المداخل متعددة المخارج التي يُفارض أنها شبه ساكنة وليست ذات انتقائية ترددية، وذلك مقارنة بشيفرة "تاروخ" المكانية-الزمانية المتعامدة المؤلفة من وحدات ذات المعدل المساوي 1/2، وشيفرة "جعفرخاني" المكانية-الزمانية المؤلفة من وحدات، وشبه المتعامدة، ذات المعدل الكامل.

وقد أظهرت نتائج المحاكاة أن الشيفرة المقترحة تُحسن الى حد كبير الأداء المتعلق بمعدل الخطأ في البتّات (BER)، في الوقت الذي تسمح فيه بمعدل إرسال أعلى بفاعلية طيفية مقدارها 4 بتّات لكل ثانية لكل هيرتز، وكشف خطي سهل باستخدام تقنيتي الكشف (ZF و MMSE).

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

USING STATIC ANALYSIS TOOLS FOR ANALYZING STUDENT BEHAVIOR IN AN INTRODUCTORY PROGRAMMING COURSE

Ibrahim Albluwi[1] and Joseph Salter[2]

(Received: 15-Mar-2020, Revised: 2-May-2020, Accepted: 18-May-2020)

ABSTRACT

Analyzing student coding data can help researchers understand how novice programmers learn and inform practitioners on how to best teach them. This work explores how using static analysis tools in programming assignments can provide insight into student behavior and performance. The use of three static analysis tools in the assignments of an introductory programming course has been analyzed. Our findings confirm previous work regarding that formatting and documentation issues are the most common issues found in student code, that this is constant regardless of major and performance in the course and that there are certain error types which are more correlated with performance. We also found that total error frequency in the course correlates with final course grade and that the presence of any kind of error in final submissions correlates with low performance on exams. Furthermore, we found females to produce less documentation and style errors than males and students who partner to produce less errors in general than students working alone. Our results also raise concerns on the use of certain metrics for assessing the difficulty of fixing errors by students.

KEYWORDS

Introductory programming, CS1, Static analysis, Automated feedback, Coding style, Gender differences.

1. INTRODUCTION

Students generate a great amount of data as they learn how to program. This data is a precious mine for understanding how they learn, what challenges they face and how they interact with tools. This work complements ongoing efforts to analyze student coding behavior by tackling a type of data that has received limited attention so far. Considerable work has been done on analyzing submission and compilation behavior, compile time errors and other issues related to the correctness of student code [1]. This work analyzes issues that do not necessarily affect correctness and that can be detected using *static analysis*; i.e., without having to execute the code, like documentation, testing and style issues.

The importance of this work derives directly from the importance of analyzing student coding behavior in general. The better we understand the types of issues students face, how they address them and how they interact with the tools that report them, the better we are able to improve these tools, adapt teaching to directly address the challenges they face and intervene early to help struggling students. More particularly, studying static analysis errors provides the following two advantages:

- It is *another source of information* beside information available from compiling and executing the code. Having an extra source of information is especially important when data is scarce.
- It provides *another dimension* for looking at novice programmers. Most issues detected by static analysis tools do not prevent the code from running correctly. Such issues are orthogonal to issues that are detected by the compiler or that cause programs to crash or produce incorrect results.

In this work, we analyze a large dataset of student submissions to programming assignments in an introductory programming course. The main goal is to answer questions about static analysis errors along the following two dimensions:

The errors:

1. Which static analysis errors are most frequent in students' code?
2. Which errors appear most in initial submissions? Which errors persist in final submissions?

1. Ibrahim Albluwi is with Princeton University, Princeton, New Jersey, USA. Email: `isma@cs.princeton.edu`
2. Joseph Salter is with Quartet Health, Caldwell, New Jersey, USA. Email: `joesalter17@gmail.com`

3. Which errors take longer to fix, and which errors take more submissions to fix?
4. Is there any relationship between the number of errors in student code on assignments and their performance on exams?

The students:

1. Are there any differences in the number and type of errors between students based on their major, prior-programming experience, gender or performance in the class?
2. Are there any differences in the number and type of errors between students who partnered on their assignments and students who worked alone?

By answering such questions, this work implicitly attempts to answer also broader questions on the suitability of using static analysis to understand the behavior of students in introductory programming courses and to predict their performance.

Note that we refer to issues flagged by the static analysis tools as "errors", although these issues do not necessarily represent "errors" in the code. They represent violations to rules that are checked by the tools and that span a wide range of issues related to formatting, documentation, testing, style and possible programming bugs.

This article is organized as follows. Section 2 discusses previous work related to this study and Section 3 describes the data, context and methodology of the conducted analysis. Section 4 reports results at the aggregate level and Section 5 reports results specific to the different student subgroups analyzed in the study. A discussion of the main results and their implications is provided in Section 6 and the limitations of the study are discussed in Section 7. Future work horizons are finally indicated in Section 8.

2. PREVIOUS WORK

2.1 Learning Analytics in Introductory CS Courses

Interest in analyzing student data in introductory programming courses is part of a more general trend of interest in learning analytics. This trend gained momentum with the increased availability of resources for storing and processing data. Generally speaking, learning analytics research aims at: (1) identifying student behavioral patterns and (2) deriving interventions that improve learning [2]. Hui and Farvolden [3] proposed a framework for how and when learning analytics can be used in the classroom and demonstrated the utility of this framework with a case study in a CS1 course. Hundhausen *et al.* [2] also proposed a process model for improving CS courses using tools integrated in the IDE that collect and analyze data and apply interventions. Ihantola *et al.* [1] performed an extensive review on learning analytics in programming courses and provided a thorough discussion of issues and challenges facing the field. One of these issues is that data is mostly private, limited in size and unique to a particular institution, which makes research reproduction and replication difficult. The problem is also amplified with the bias many researchers in the CS education community have against replicating, reproducing and repeating previously published studies, as reported by Ahadi *et al.* [4].

2.2 Analysis of Coding Behavior

There is a wide range of student activity data that could be analyzed. The most relevant data to this work is coding behavior data. The most known tool for automatically recording such data is BlackBox [5], which records program line-edits in the BlueJ IDE. Other examples of tools include online coding environments like PCRS [6] and CloudCoder [7] and IDE plugins like TestMyCode [8].

Several works have analyzed the process through which students work on their assignments. For example, Piech *et al.* [9] tracked the evolution of student code over compilation events and developed a model for predicting struggling students. Karavirta *et al.* [10] classified students into categories based on their pattern of resubmission to an automatic grader. They found that some students used the ability to resubmit in an inefficient way and that limiting the number of submissions for such students at the beginning of the course could improve their performance later on. Blikstein [11] found that novice students tended to copy and adapt large batches of sample code, whereas experienced students tended to code more incrementally. Blikstein *et al.* [12] also found that changes in code update strategies over the semester were correlated with performance on exams. On the other hand, Allevato

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

and Edwards [13] found that students who had more increasing changes in code between submissions to an automatic grader generally performed better in the course.

Several works have also analyzed errors that novice programmers make. For example, Altadmri and Brown [14] studied 37 million compilation events to analyze the frequencies of different errors and the time taken to fix them. Brown and Altadmri [15] also found that instructor beliefs about error frequencies do not agree with the data. Tabanao *et al.* [16] used compile-time errors and Jadud's Error Quotient [17] to detect at-risk students and predict midterm exam scores, which is an algorithm that has been shown to find correlation between errors and student performance. It is difficult though to tell whether or not it can be used as a reliable predictor [18]-[19].

2.3 Static Analysis Tools

Static analysis tools are useful for detecting code that does not follow programming standards and pointing out potential bugs, performance issues and flaws in logic. The tools used in this study are Checkstyle[1], PMD[2] and FindBugs [20], which are among the most commonly used open source static analysis tools. Checkstyle has a focus on Java style. PMD focuses more on programming flaws and unreadable or non-simplified code than on style. Common PMD checks include finding unused variables, non-simplified expressions and empty statements. FindBugs is run on byte code rather than on source code. Its purpose is to find byte patterns that are often bugs, like assignments to variables that are never used and values that will always be null. Together, Checkstyle, PMD and FindBugs search for a wide range of flaws in programs, both cosmetic and non-cosmetic.

Although these tools can be used for educational purposes, they were originally designed for software professionals. Therefore, much of the research on these tools targets the non-educational community (e.g. [21]-[24]). In an early study, Mengel and Yerramilli [25] argued for the value of using static analysis tools intended for professionals in grading student programs. Nutbrown and Higgins [26] warned against applying direct mark deductions based on errors reported by these tools. They found this to produce results that are very different from how instructors evaluate student code. Truong *et al.* [27] developed a static analysis framework specifically for checking the quality of student programs and their structural similarity to model solutions. Their goal was to provide better-than-standard feedback to students and tips for improvement. Other similar tools were also described in the literature, like PyTA [28], Gauntlet [29] and Expresso [30]. For reviews on how static analysis tools can be used for educational purposes, see Striewe and Goedicke [31] and Rahman and Nordin [32].

2.4 Static Analysis of Student Code

Edwards *et al.* [33] conducted a thorough qualitative study of issues reported by CheckStyle and PMD for around half a million student Java submissions. They found that documentation and formatting issues were the most commonly reported issues by the tools and that issues that are potentially coding flaws were most indicative of performance even when students fixed these issues. They also found that the most common issues were consistently the same among majors, non-majors and students at different experience levels. In our work, we conduct the analysis along the same lines in Edwards *et al.* [33]. Our results confirm the above-mentioned results, with a few differences as will be discussed in Section 6. Our work also expands the analysis, reports new results and provides further insights.

Keuning *et al.* [34] conducted an analysis of errors reported by PMD on student code. However, their analysis was restricted to issues related to code quality, like modularization, decomposition and the use of idioms, which is a subset of what static analysis tools report. Their analysis included issue frequencies and time needed to fix them. Their work also compared students who used static analysis tools to students who did not and concluded that quality issues in student code are rarely fixed and that students typically ignore issues reported by the tools. These results seem to be course-specific, as they are not consistent with what has been observed in this work as will be discussed in Section 6. Liu and Peterson [28] also reported positive results when using PyTA in an introductory course, where they observed (compared to a previous year) a significant reduction in the number of repeated errors per submission, submissions to pass a programming exercise and submissions required to solve the most

1 https://checkstyle.sourceforge.io/
2 https://pmd.github.io/

common errors. Other studies confirming the utility of using static analysis tools in introductory programming classes include Delev and Gjorgjevikj [35] (in C) and Schorsch [36] (in Pascal).

3. METHODOLOGY

3.1 Context

COS 126 is the introductory programming course at Princeton University. The course is required for CS majors and engineers and open for non-majors. The course has 9 Java programming assignments (46% of the course grade) and a machine code assignment (4% of the course grade). Students are allowed to work with a partner on the last 4 of the Java assignments.

On assignments where partnering is allowed, students are required to follow the pair programming protocol [37]. The collaboration policy on the course website provides the details of this protocol and instructors emphasize verbally in their sections that students who partner must follow it. Students who partner must also state in a readme file submitted with each assignment that they have followed the pair programming protocol. However, there is no way to tell whether or not students actually followed the pair programming protocol beyond taking their word for it.

Students submit their work to an online system that runs automated tests for checking correctness, performance and API adherence. The system also runs CheckStyle, PMD and FindBugs and includes issues reported by these tools in the automated feedback. Students are allowed to receive feedback from the system as many times as they wish before marking their submission as ready for grading. Human graders then use a rubric to grade the final submissions and provide personalized feedback on correctness, performance and style. The feedback in the first assignment includes a warning for students who do not address all the issues flagged by the tools and indicates that a deduction will be applied in later assignments. These deductions are minor (typically ≤ 3 points out of 20) and depend on how many issues flagged by the static analysis tools are left unaddressed. The rubrics are not provided to the students beforehand, so it is not directly clear to them what the deductions are before their work is graded. Note that students are not directly "taught" how to format or comment their code in the classroom but are provided with a link to a page that lists the style guidelines for the course.

The IDE used in the course was Dr. Java, which does not automatically format the code, but provides a menu option for achieving that. While the course did not require using Dr. Java, it did not provide support for other IDEs and used a custom version of the IDE pre-packaged with the libraries required in the course. This made almost all students use it instead of other IDEs. Anecdotally, the number of students who used IDEs other than Dr. Java was very small and insignificant.

The course has two programming exams (7.5% each) and two written exams (17.5% each). Programming exams are conducted on-campus under exam conditions, where students are expected to complete within 80 minutes a few programming exercises that are much smaller in size than the programming assignments. Grading is done based on code correctness only, without any consideration to style, commenting or testing issues. Written exams are also timed (80 minutes each) and conducted on-campus under exam conditions. However, they test programming knowledge as well as other computer science concepts, like algorithm analysis, digital logic and theory of computation (state machines, computability and intractability). Programming knowledge in these written exams is tested with questions that do not involve code writing (e.g. multiple-choice, true/false, …etc.)

The course has an average score of 87.2% and a standard deviation of 7.2%, where around 0.6% receive an F grade at the end of the semester and around 5.3% drop out after the first written exam. Students who drop out in the first few weeks of the semester (before the first exam) are typically students who "shop" for courses rather than students who struggle with the material. Moreover, not all students who drop out after the first exam are struggling students, as it is not uncommon for some students to drop the course if they feel that they won't achieve a grade of an A or an A-.

3.2 Dataset

Data has been collected in COS 126 in three semesters; Fall 2016, Spring 2017 and Fall 2017. The data includes every Java file in every submission for every student who attended the final exam of the course in these three semesters. This includes intermediate submissions that were not considered for grading. The data also includes the text of every piece of feedback students received from the automated feedback system, as well as timestamps for when the files were submitted. Overall, the dataset has a total of 1,051,105 occurrences of 304 distinct issues flagged by the static analysis tools covering a total of 968 students who completed the course. This is around 87.8% of the students who were enrolled in the course on the first day of classes in each of the three semesters. Around 13.2% of the students dropped out at different points in the semester and were excluded from the analysis, as it is impossible to tell for each of them until which point in the semester they were taking the course seriously. As discussed before, a large proportion of these students did not sign up for the course with the intention of completing it.

3.3 Metrics

The analysis in this work broadly follows the analysis performed by Edwards *et al.* [33]. The first two of the metrics used in the analysis are commonly used in error analysis studies and were implemented in the same way as reported by Edwards *et al.* [33]. The third metric was not used by Edwards *et al.*

- ***Error Frequency***. Error counts for each source file were normalized by the number of lines of code and considered as occurrences per thousand lines of code (KLOC).
- ***Time-to-Fix***. This is an estimate for how long an error stays unfixed in the student's code after it has been reported by the tool. An increase in the frequency count of an error from a submission to another is considered as an introduction of new errors, whereas a decrease in the count is considered as a resolution of errors. The *time-to-fix* is the difference between timestamps of introduction and resolution events for a certain error.
- ***Submissions-to-Fix***. This is an estimate for the number of submissions taken by the student to fix the error. The same protocol used for computing the time-to-fix is used, but the submission number is recorded instead of the timestamp.

Note that the time-to-fix is not necessarily the same as the actual time-on-task by the student trying to fix the error. The student might take breaks between submissions or put off fixing the error if they feel that it is not important to fix the error immediately.

3.4 Error Categories

Static analysis tools report many errors on different types of issues. Edwards *et al.* [33] grouped these errors into categories that we adopt in this work:

- ***Coding Flaws***: "Constructs that are almost certainly bugs (such as checking for null after a pointer is used ...)".
- ***Excessive Coding***: "Size issues, such as methods, classes or parameter lists that exceed the expected limits and may indicate readability problems".
- ***Formatting***: "Incorrect indentation or missing whitespace".
- ***Naming***: "Names that violate capitalization conventions, are too short or are not meaningful".
- ***Readability***: "Issues other than formatting that reduce the readability of the code".
- ***Style***: "Code that can be simplified or that does not follow common idioms".

The following categories were adopted with modification:

- ***Documentation***: Unlike [33], Checkstyle Javadoc comment checkers are ignored, since Javadoc comments are not taught in the course. The issues considered instead are: not preceding a field or a method by a comment, leaving an empty method body without a comment, not adding a header comment for a file and not formatting a header comment according to the course guidelines.
- ***Testing***: Formal unit testing was not required in the course. Therefore, this category includes only one custom CheckStyle error that checks whether or not the student calls all public methods in the main method. Understandably, this is a naive check. A student might call a public method in main, but not genuinely test it (by printing out the result, for example). The check does not catch such instances, but submissions flagged by the check certainly lack sufficient testing.

Table 1. Examples for each of the 10 most frequently occurring errors.

Error	Category	Example
`WhitespaceAround`	Formatting	`x=4` instead of `x = 4`.
`Comment`	Documentation	A method is not preceded by a comment.
`RegexpSingleline`	Documentation	The student name or ID is missing from the header comment.
`IllegalTokenText`	Formatting	`/*comment*/` instead of `/* comment */` or `//comment` instead of `// comment`
`MainCallsAllPublicMethods`	Testing	A public method is not called in main
`WhitespaceAfter`	Formatting	`(int)x` instead of `(int) x`.
`RegexpMultiline`	Style	Not starting every line in a multiline comment block with an asterisk.
`LineLength`	Formatting	A line has more than 85 characters.
`NumericLiteral`	Style	Using a numeric literal instead of defining and using a constant.
`RegexpHeader`	Documentation	Header comment is missing from a file.

Edwards *et al.* [33] also include a Braces category for errors related to missing optional braces. This category was omitted, because such errors were not encountered in our analysis. It must also be noted that Edwards *et al.*'s analysis includes 112 distinct errors across Checkstyle and PMD, whereas ours included 304 distinct errors across Checkstyle, PMD and FindBugs. Errors that were not seen by Edwards *et al.* were categorized into the above groupings[3].

4. RESULTS: AGGREGATE STUDENT ANALYSIS

We first performed an analysis on average error statistics across all students to describe the behavior of a typical student in the course.

4.1 Error Category Frequency

After grouping errors into the categories described in Section 3.4, the resulting frequencies are displayed in Figure 1. The figure shows the average error frequency across all submissions compared to error frequencies in the initial and final submissions. To put the frequency rates per KLOC into perspective, student assignments in this study contained an average of 194 lines of code, which means that an average submission has around 6 errors in the Formatting category, 3.7 in the Documentation category and 2.4 in the Style category.

Considering the average error frequencies, Formatting errors are most prevalent, with an average of 31.24 per KLOC, followed by Documentation errors at 19.16 per KLOC. Together, Formatting and Documentation errors account for 60% of the total errors in the student code. Note that Formatting and Documentation errors encompass only 19 of the 304 unique error types used in the analysis. So, 60% of the error count is covered by just 6.25% of the error types seen. Appending Style to this count adds 121 unique error types and 15% to the total error count. The top three error categories, then, account for 75% occurrences of all errors. The vast majority of errors being made are thus limited to a small number of different error types.

Looking at final submissions, it is clear that *most* of the errors get fixed, with Documentation errors getting fixed at a very high rate. Overall, only 2.3% of the total errors occurred on final submissions. Formatting errors remain by far the most common, which could be the result of formatting error messages being less clear to novices compared to error message on missing comments. It could also be that students are less receptive to changing their style in formatting the code. The small discrepancy between Testing errors on initial *versus* average submissions suggests that students might be adding tests only in their final submissions to silence the error, as opposed to writing tests to actually test their code. More on this issue will be discussed in Section 6.

3 A full listing of the errors along with their groupings and occurrence per KLOC can be found at: https://figshare.com/s/cbe48ead7dcc7b15a5bf

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

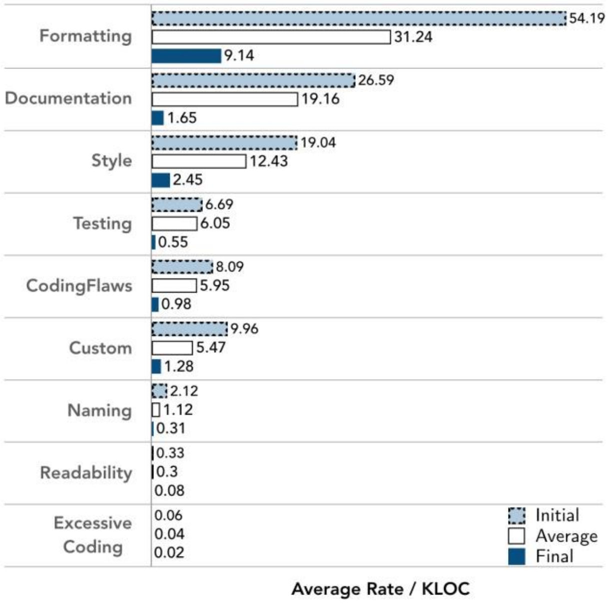


Figure 1. Error rates (in KLOC) on initial, average and then final submissions.

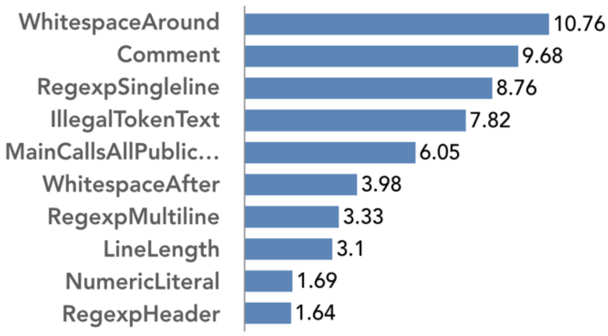


Figure 2. Average rate per KLOC for the 10 most frequently occurring errors.

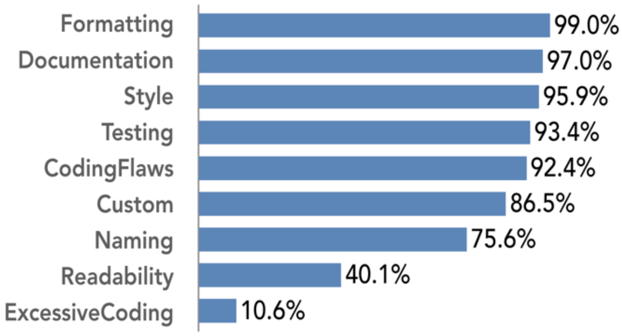


Figure 3. Percentage of students who make errors from each category.

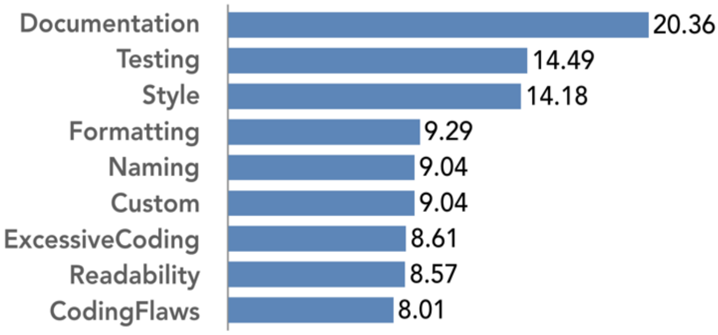


Figure 4. Average number of hours taken to fix errors from each category.

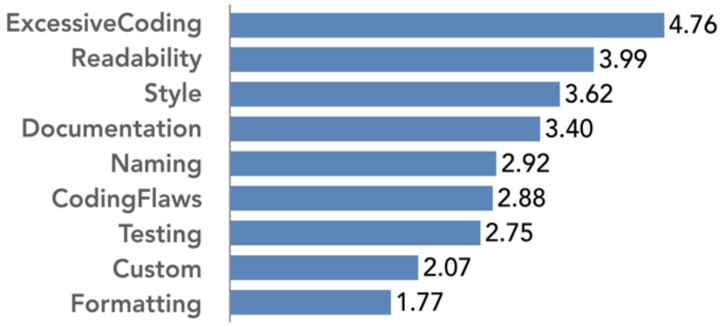


Figure 5. Average number of submissions taken to fix errors from each category.

4.2 Error Frequency and Popularity

Considering the errors individually, Figure 2 shows the 10 most frequently occurring errors, where Formatting and Documentation errors account for seven of the top 10. Table 1 shows examples for each of these errors to explain what they mean. Figure 3 shows the percentage of students who make errors from each category at some point during the semester. Almost all students make Formatting, Style, Documentation and Coding Flaw errors, while slightly less make Testing and Naming errors. Readability and Excessive Coding errors are much less common. This order closely resembles the order of frequency presented in Figure 1. We will describe later in more detail which types of students produce more errors from each category.

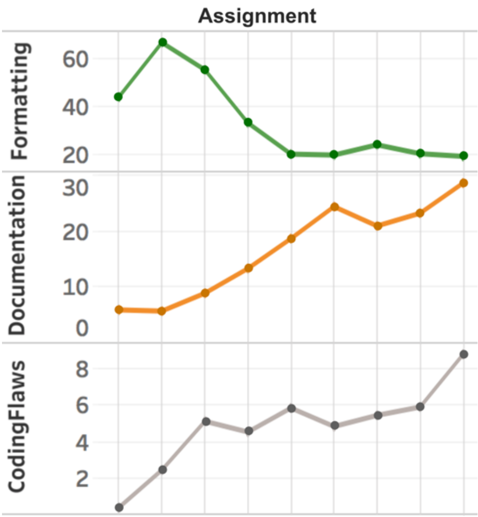


Figure 6. Error rates per KLOC over time for assignments 1 to 9.

Table 2. Average exam scores for students who leave at least one error in a final submission for any assignment *versus* those who fix all errors.

	Written Exam		Programming Exam	
	Errors	No Errors	Errors	No Errors
CodingFlaws	69.4%	76.8%	79.3%	87.4%
Testing	67.0%	75.8%	78.0%	86.0%
Style	72.2%	77.0%	83.0%	86.7%
Documentation	68.8%	76.3%	79.5%	86.5%
Formatting	72.4%	76.7%	83.1%	86.5%

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

4.3 Error Progression over Time

Figure 6 displays the average rates of the error categories that show an interesting trend line. Error categories not shown have arbitrarily fluctuating trend lines or error rates that are too small for the trend line to be meaningful. The clear decline in Formatting errors suggests that students eventually gain a grasp on what their code should *look* like. This is interesting, as the only instruction students receive in the course on formatting is through the error messages produced by the static analysis tools. Another possible explanation for this decrease in Formatting errors is that more students into the semester might start using the code formatting option in Dr. Java, which is hidden in one of the menus. However, this is difficult to tell from the data.

On the other hand, Figure 6 shows an increase in Documentation and Coding Flaw errors over the semester. The increase in Documentation errors could be explained by lack of care; over time, students may grow tired of Documentation errors and choose to ignore fixing them until the final submissions. The increase in both the Documentation and Coding Flaw errors could also be related to the growing complexity of assignments over the semester.

4.4 Correlation with Exam Performance

We examined the total error counts, as well as the error counts from each error category, *versus* grades in programming exams and written exams. No apparent correlation was revealed in any case. However, we found a correlation between the *presence* of error categories in final assignment submissions on both written and programming exam performance. Table 2 displays average written exam scores for students who left at least one error from the given category in a final submission for any assignment *versus* those who did not. Only error categories with statistically significant differences are shown (T-test p-val. < 0.0001). For reference, the average score on written exams was 74.2% with a standard deviation of 12.8% and on programming exams was 84.7% with a standard deviation of 12.9%.

5. RESULTS: STUDENT SUBGROUP ANALYSIS

5.1 Low-Performing *versus* High-Performing Students

We have already examined the relationship between error frequencies and exam performance. We provide here a more detailed analysis to compare low- and high-performing students in the course in general. To perform this analysis, we break students into groups. We approached this in the following three different ways:

1. Break students into three evenly sized tertiles based on their final course average.
2. Create a C+ and-below subgroup, a B-range subgroup and an A-range subgroup based on final course average, using the standard cutoff points (80% for B; 90% for A).
3. Group students by taking course averages of at least 1 standard deviation below the mean, within one standard deviation of the mean and at least one standard deviation above the mean.

It turned out that the three options listed above produced very similar results in terms of significant differences in error behavior between subgroups. Consequently, we choose to present Option 1 and we define the three subgroups as low-performers, medium-performers and high-performers. For reference, the mean course grade was 87.2% with a standard deviation of 7.2%.

Table 3 shows statistics for each subgroup individually compared to all the students together. The “Avg. Duration" column is the average duration between initial and final submissions. Note that the “Avg. Duration" does not necessarily reflect the actual number of hours that a student spent working on the assignment. It describes the duration between the first and last files submitted.

As seen in the table, low-performers produce more errors in overall than the average student, while high-performers produce significantly less. A one-way ANOVA shows that the error rates for each performance group are statistically significant ($F = 42.8$, $p < 0.0001$) and Tukey’s HSD test[4] shows that the average error rates for low-, medium- and high-performers are statistically different.

4 Tukey’s HSD is a statistical tool often used as a *post hoc* test with ANOVA. ANOVA provides a significance test; Tukey’s HSD compares all pairs of means and determines which pairs are statistically different.

Table 3. Student performance groups.

Performance Level	Grade Range	Number of Students	Errors/ KLOC *	Average Duration *	Average # of Submissions **	Hours to Fix **	Submissions to Fix **
Low	47% - 85.9%	329	88.2	41.2 hrs.	16.7	12.6	5.14
Medium	85.9% - 91.1%	318	75.2	54.0 hrs.	17.4	15.6	5.05
High	91.1% - 99.2%	321	66.6	63.5 hrs.	14.4	15.25	4.3
All Students	47% - 99.2%	968	74.3	52.8 hrs.	16.17	14.5	4.85

*: Significant differences between all groups.
**: Significant differences between high- and low-performers and between high- and medium-performers.
Calculated by ANOVA and Tukey's HSD.

The table also shows that low-performers, interestingly, take less time, but more submissions, to fix errors compared to high-performers and these differences are significant (time-to-fix: $F = 36.7$, $p < 0.0001$; submissions-to-fix: $F = 118.1$, $p < 0.0001$) and different from each other, as suggested by Tukey's HSD. This may suggest that low-performers submit their code more blindly than high-performers and cram their work into shorter periods of time, which requires them to submit more and at a faster rate. On the other hand, high-performers seem to rely less on the automatic grader, spending more time in overall on assignments while using less submissions.

Figure 7 displays the difference in average rate per assignment across error categories for each performance group, omitting the statistically insignificant categories (Naming, Readability and Excessive Coding). It is evident that high-performers produce less errors in all categories than medium- and low-performers and medium-performers produce less errors than low-performers in all categories but Documentation and Testing. An ANOVA suggests that the most significant differences among the three are in Formatting and Coding Flaws and the least significant differences are in Documentation and Testing. The disparity in Coding Flaws suggests a higher level of misunderstanding amongst low performers. Interestingly, all three performance groups seem to be testing their code relatively equally.

Further analysis reveals that differences in error counts on *initial* submissions are statistically insignificant for all categories except for Coding Flaws. Low-performers produce 11.5 of these types of errors per KLOC in initial submissions, compared to 8.92 from medium-performers and just 5.79 for high-performers ($F = 23.5$, $p < 0.0001$). The course average is 8.75. This suggests that low-performers are producing more buggy code on their initial submissions.

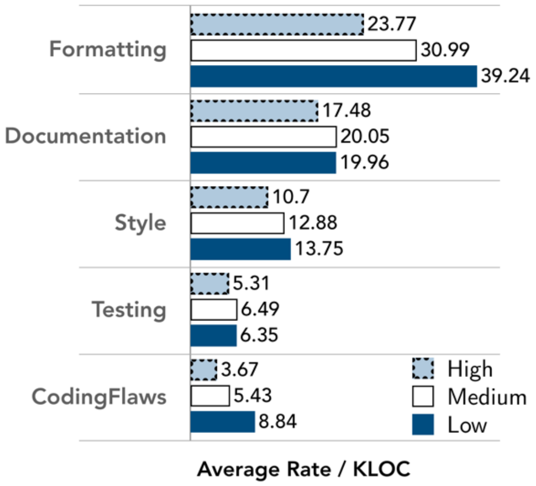


Figure 7. Error category rate per KLOC for different performance groups.

Table 4. Student major groups.

Major Group	Number of Students	*Avg. Course Grade	**Avg. Duration	***Avg. # of Submissions
CS	97	89.7%	55.4 hrs.	14.8
Engineering	230	86.6%	46.3 hrs.	15.9
Other Science/Math	40	87.8%	54.5 hrs.	13.9
Social Science	46	84.2%	47.9 hrs.	15.0
Humanities	555	87.4 %	56.0 hrs.	16.8

* Significant differences between CS and all other majors except Science/Math.
** Significant differences between Engineering and Humanities.
*** Significant differences between (CS and Humanities) and (Humanities and Other Sciences).
Calculated by ANOVA and Tukey's HSD.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

5.2 Majors *versus* Non-Majors

Students were grouped into five major groups, as shown in Table 4. Analyzing error counts per KLOC and the number of submissions taken to *fix errors* (not shown in the table), we found no significant differences between the major groups. However, we found CS students to take significantly longer to fix errors than each of the other groups (17.6 hours per error). Non-CS Engineering students take just 12.7 hours, which is significantly shorter than humanities and social science students (14.8 hours). This difference in the time-to-fix, but not in the submissions-to-fix, suggests that CS majors might be working longer before submitting or taking longer breaks between submissions. Engineers on the contrary seem to be cramming their work into a shorter amount of time. This contrast between CS majors and engineers seems difficult to explain only using the data.

We also performed three binary comparisons: first grouping CS and engineering students together and comparing them to their counterpart and second comparing students in STEM fields *versus* those not in STEM fields. These yielded no significant results. However, comparing CS majors and non-CS majors reaffirmed our earlier statement that CS majors take more time to fix errors ($F = 30.2$, $p < 0.0001$). It also showed that CS majors produce less Coding Flaw errors on initial submissions, averaging 5.9 occurrences per KLOC *versus* 9.1 from non-majors ($F = 5.8$, $p = 0.003$).

5.3 Students with Different Prior Programming Experience

Students in the course entry survey indicated their level of programming experience by choosing between “None”, “Some”, “Plenty” and “I’ve worked on some sizable projects”. Omitting the latter two categories due to their small size, Table 5 contains the description of the former two.

Table 5. Comparison between student groups.

	# of Students	Avg. Grade*	Avg. Duration*	Avg. # of Submissions
No Prior Experience	433	85.1%	49.6 hrs.	16.5
Some Prior Experience	458	88.7%	56.6 hrs.	16.0
Male	545	87.8%	53.1 hrs.	16.1
Female	390	86.7%	53.2 hrs.	16.2

* Significant differences between students with some and with no prior programming experience (ANOVA).

Table 6. Male *vs.* female characterization.

	Male	Female
% Majoring in CS	11.7%	7.95%
% Majoring in Engineering	24.8%	19.7%
% Majoring in Humanities	54.1%	64.6%
Avg. Prior Experience	0.69	0.56

Table 7. Differences in male *vs.* female error frequencies on initial, average and final submissions.

Gender	Testing			Style			Documentation		
	Initial	Average	Final	Initial	Average	Final	Initial	Average	Final
Female	6.46	Insignificant	Insignificant	17.5	10.7	1.76	24.0	15.9	0.91
Male	7.73	Insignificant	Insignificant	21.6	13.6	2.77	32.5	21.3	2.09

Table 8. Average partner statistics.

	Course Grade	Duration*	Number of Submissions*
Pair	87.2%	54.4 hrs.	17.7
Solo	87.9%	97.4 hrs.	24.9

* Significantly different (ANOVA).

Table 9. Significant differences in error rates on initial submission.

	Coding Flaws	Testing	Style	Doc	Formatting	Naming
Pair	8.45	9.2	17.3	32.1	22.3	1.3
Solo	12.0	12.5	25.6	43.2	32.6	2.2

Students with no prior programming experience and those with some show no statistically significant differences in their error rate or time-to-fix. An ANOVA on average number of submissions-to-fix does yield significant results (F = 12.12, p < 0.0001), but the submission numbers are very close together to have any meaning. Breaking down the errors into categories, we again find one significant result regarding Coding Flaw errors on initial submissions. While the average student produces 8.75 of this error per KLOC in their initial submissions, students with no experience generate an average of 10.6 and students with some experience generate 7.61 (F = 12.7, p < 0.0001). This shows a repeated trend with Coding Flaw errors on initial submissions. Aside from this finding, though, prior programming experience does not seem to be clearly related to the static analysis error behavior.

5.4 Gender Differences

The gender breakdown of the students is displayed in Table 5 and Table 6 attempts to characterize the typical male *versus* the typical female in our course. There are more females in the humanities than males and more males in CS and Engineering than females. Male students also typically enter the course with more prior programming experience than female students. The averages in Table 6 (0.69 and 0.56) were computed by considering “no prior experience” a 0 and “some prior experience” a 1.

An analysis of the aggregate error counts showed that female students produce an average of 67.8 static analysis errors per KLOC on assignments, while males produce 78.3 (F = 13.0, p < 0.0001). Females also take slightly less time to fix errors – 13.5 hours *versus* 15.2 for males – and also slightly less submissions – 4.7 *versus* 5 (time-to-fix: F = 21.8, p < 0.0001; submissions-to-fix: F = 27.5, p < 0.0001). The differences in these figures are not enough to justify a deep analysis.

Breaking down error frequencies based on error categories, Table 7 displays the significant differences found in initial, average and final submissions between male and female students. As the table shows, female students produce less *Style* and *Documentation* issues on all submissions. Female students also test their code more on *initial* submissions, but the differences eventually even out in later submissions. These differences were not seen when comparing students by major or by prior experience, which suggests that they are gender related.

5.5 Pair *versus* Solo Programmers

To test whether students who work with a partner produce less errors than students who work individually, we limited our analysis to only assignments where students were given the option to partner (the final four assignments in the course). This ensures that, when comparing partnering *versus* non-partnering students, all students in the analysis were working on assignments of equal difficulty. Note that on average, around 43% of the students worked individually in the assignments we analyzed.

Table 8 shows statistics on partnering *versus* non-partnering students in assignments where partnering was allowed. Immediately clear is the significantly fewer hours and submissions spent on assignments by students who partner. On an aggregate level, students who partner produce 61.4 errors per KLOC per assignment on average, compared to 77.3 for those working alone (F = 19.2, p < 0.0001). Partnering students also take 15.9 hours to fix errors on average compared to 19.3 for non-partnering students (F = 29.6, p < 0.0001). In terms of the number of submissions taken to fix errors, partnering students take slightly more than non-partnering students, but the difference is too small to report.

Partnering students also make fewer errors across all categories. Table 9 displays the differences for error categories that yielded significant results. The table shows that partners make significantly less errors on initial submissions for both non-cosmetic and cosmetic issues. The fact that partnering students produce less Coding Flaw and Testing issues on initial submissions suggests that students working in pairs are able to notice potential bugs more efficiently and are keener on testing their code early on than students working alone. The variance in the remaining error categories suggests that partnering students help each other conform to style standards. It is also interesting to note that differences in error rates eventually even out in final submissions. This suggests that students working individually are solving their errors by the end of the submission process. But, because they spend more time and more submissions on assignments, they are working harder to do so.

6. DISCUSSION

The previous sections report many positive and negative results that vary in their significance. In this section, we summarize and discuss some of the main findings.

Effectiveness of Feedback from Static Analysis Tools. It is clear from Figure 1 that students fix most of the issues reported by the static analysis tools, where the number of errors drops drastically from initial to final submissions. There is also evidence that students might be learning from some of these error messages, as the number of formatting errors drops as students progress in the semester (see Section 4.3). This is consistent with what is reported by Edwards *et al.* [33], but not with what is reported by Keuning *et al.* [34]. Since there were rubric deductions in this study and in Edwards *et al.*'s study for not addressing issues raised by the static analysis tools (but not in Keuning *et al.* [34]), students might be fixing errors reported by the tools, because they want to avoid deductions. However, if these tools were not present, it is not clear whether or not deductions alone would have been enough for students to fix the issues!

Gender Differences. Results in Section 5.4 show that female students make a statistically significant lower number of errors compared to male students in the Documentation and Style categories in initial, intermediate and final submissions. Such differences were not spotted with other student groups. For example, high-performing students, students with more programming experience and students who partner make fewer errors in *all* categories. Moreover, low-performing students and students who do not partner produce more errors in *initial* submissions in the *Coding Flaws* category. This strengthens the hypothesis that female students in the course are more careful with writing comments and code that follows recommended style.

Up to our knowledge, this result is new. Previous work pointed out gender differences in learning programming [38]-[39], debugging [40], interacting with software [41]-[42], performance in CS degrees [43], self-efficacy in programming [44] and programming contests [45]. However, we are not aware of studies showing the presence or absence of gender differences in coding style.

While the result fits a stereotypical view of females being more concerned about aesthetics than males, we should be careful when interpreting the results. For example, the results do not imply that there is a certain programming style distinct to females, nor do they necessarily imply that style is a good predictor of gender. In a study by Carter and Jenkins [46], instructors were asked to identify the gender of students by examining pieces of code randomly selected from student solutions to assignments. Results revealed that the aforementioned stereotype was clearly present in how the instructors attempted to identify the gender of the students. However, empirical results supporting this stereotype (beyond the results reported here) are still absent.

Effect of Partnering. Our results show that students who partnered produced fewer static analysis errors, completed their assignments faster and submitted less to the automatic grader. This result is consistent with the literature on pair programming in introductory programming courses (*e.g.* [47] and [48]). However, the results reported here are different, since they relate to code quality issues that do not affect correctness, while most previous work measured code quality by the number of passed test cases or exam and assignment grades [49]-[50]. As mentioned in Section 3.1, it is important to note that the pair programming protocol was required for students who worked in pairs, but there was no way to enforce it, as students worked outside the classroom environment.

Commenting Behavior. Results showed that Documentation errors are the second most common error type. They stay un-fixed longer than any other type and students receive more of them as the semester proceeds. This is equally the same regardless of the student major, performance in the course or prior programming experience. The only difference was with female students who produced a statistically significant lower number of Documentation errors than males. This pattern of ignoring Documentation errors is also consistent with the results reported by Edwards *et al*. [33]. What is interesting is that Javadoc checks were ignored in this study, while they were the main measure for Documentation in the study of Edwards *et al*. and yet the results were the same. This suggests that these results relate to writing comments in general.

What is interesting about this result is not that the number of Documentation errors is high, as this could be explained by the fact that a distinct error is generated for every missing comment for a field or a method, which means that a student who chooses to comment his/her code at the end would receive *many* Documentation errors for every submission he/she makes except for the last one. What is interesting about the results is that students delay writing comments until the end. It is difficult to tell whether or not this behavior is because students add comments only to avoid mark deductions or because they believe that it is unimportant to write comments *while* coding. Hence, an implication of these results is that instruction should directly emphasize the importance of commenting in general and that instructors who wish their students to develop the habit of commenting *while* coding should explicitly teach them to do so.

Testing Behavior. Although the assignment instructions required a very simplistic form of testing, the data shows that students seemed to avoid writing such tests until the very end (Section 4.1), most probably to avoid losing the testing points. The results also show that both low- and high-performing students produced a similar amount of testing errors, even though high-performing students produced fewer errors in overall (Section 5.1).

This behavior is likely the result of allowing students to receive feedback from the automatic grader without limits, which made it easier for students to use the automated tests than write tests on their own. Moreover, the course did not teach testing and did not teach or require test-driven development, which could explain why although the same unlimited submission policy is used in [33], their results on testing do not agree with ours. In fact, Edwards argued for teaching Test-Driven Development early in the curriculum to move students away from trial-and-error programming [51]. Another way around this behavior, suggested by Karavirta *et al.* [10], that does not involve formally requiring unit testing or teaching it, is to limit the number of allowed submissions to the automatic grader in the first few assignments, so that students could develop better testing habits at the beginning of the semester.

Association between Performance and Static Analysis Errors. Results showed an association between static analysis errors and performance in two ways:

1. The mere *presence* of errors in *final* submissions was found to be associated with lower performance on *exams* (see Section 4.4). This is interesting, because these errors are not directly related to the material students are tested on during exams. One possible explanation for this result is that students who are content with leaving issues in their final assignment submissions (despite seeing the automated feedback) are lazier students in overall, whose laziness shows also as weaker performance on exams compared to students who make sure to address all the automated feedback they receive. Another plausible explanation is that students who are unable to address all the issues by their final submission are students struggling with programming in overall.
2. Lower-performing students, non-majors and students with less prior programming experience were found all to have *more* errors in their *initial* submissions from the Coding Flaws category.

This suggests that these two indicators are good candidates for further investigation on their ability to predict at-risk students and the possibility of their use as features in machine learning models (see Hellas *et al*. [52] for a review on predicting academic performance in computing education).

7. LIMITATIONS

One of the goals for using the time-to-fix and submissions-to-fix measures was to investigate which errors students struggle with most. These measures have also been used by other researchers for the same purpose (*e.g.* [53] and [14]). However, inferring meaning from results using these measures proved to be tricky. Taking longer time or more submissions to fix an error can plausibly be attributed to the student ignoring the error instead of not being able to fix it. The data can also be distorted by students work habits. For example, students who start early or take breaks while working on the assignment will show longer time-to-fix values than students who complete their work in one sitting. Also, students who tend to rely more on the automatic grader will show higher submissions-to-fix values regardless of whether these submissions were actually attempts to fix the errors.

Results reported by Edwards *et al.* [33] for error categories that take longer to fix were actually almost the reverse of ours. This shows that these measures could be highly sensitive to course-specific factors, like how assignments are graded, what instructions are given to students and what material is emphasized more. These shortcomings of the measure should be kept in mind by researchers using it. For more discussion of time metrics used in the analysis of student programming behavior, see Leinonen *et al.* [54].

The variance in student working habits and submission strategies affects also how error rates per KLOC should be interpreted. As shown in Section 4.1, the vast majority of errors belong to the Formatting, Documentation and Style categories, which students might be ignoring until their last submission, thus compounding the total number of errors. Therefore, more emphasis should be placed on Coding Flaw errors, as they could reflect bugs and misunderstandings that need direct attention. As Edwards *et al.* [33] note, these errors are often masked by the abundance of cosmetic errors which students place less emphasis on. It might be useful to think about how stronger emphasis could be placed on Coding Flaw errors in automated feedback systems to make students keener on fixing them.

Another limitation of this work stems to emanate from how errors were grouped into categories. It is reasonable to expect the results to be dependent on which error groups were used in the study and which errors were assigned to each category. Also, the fact that students received feedback from the static analysis tools *only* through the submission system, rather than by directly running the tools or having them embedded in the IDE, might have affected when and how the students fixed the errors. Finally, this study was conducted at a highly selective school (acceptance rate < 7%), which means that the average student in this study might be academically stronger and more motivated than the average student at other schools. This is clear from the low failure and drop-out rates described in Section 3.1 when compared to the failure rates reported at other institutions [55]. The analysis also excluded students who dropped out from the course (see Section 3.1 and Section 3.2), which must have left some of the lower-performing students unconsidered.

8. Future Work

While performing the analysis on error categories has provided many insights, future work could benefit from studying individual errors instead of broad categories. Looking at individual errors can provide more fine-grained information on the types of difficulties and misconceptions novices face. This is especially true for errors in the Coding Flaws and Style categories. Errors associated with performance in the course and errors that appear most commonly in student code should be addressed explicitly by instructors in their teaching.

More work can also be done to investigate the effect of different programming languages on the errors students make and their ability to resolve them. Another interesting direction of research is to study how errors reported by the tools can be accounted for in rubrics used by automatic graders and how these tools can report issues in a way that emphasizes potentially serious issues more than other less important issues. This could affect how student respond to the error messages they see.

Finally, more work needs to be done on how students perceive commenting and to better understand the relationship between gender, pair programming and coding style.

References

[1] P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers and others, "Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies," Proceedings of the 2015 ITiCSE on Working Group Reports, pp. 41-63, [Online], Available: https://doi.org/10.1145/2858796.2858798, 2015.

[2] C. D. Hundhausen, D. M. Olivares and A. S. Carter, "IDE-based Learning Analytics for Computing Education: A Process Model, Critical Review and Research Agenda," ACM Transactions on Education (TOCE), vol. 17, no. 3, pp. 1-26, 2017.

[3] B. Hui and S. Farvolden, "How Can Learning Analytics Improve a Course?," Proceedings of the 22nd Western Canadian Conference on Computing Education (WCCCE'17), pp. 1-6, [Online], Available: https://doi.org/10.1145/3085585.3085586, 2017.

[4] A. Ahadi, A. Hellas, P. Ihantola, A. Korhonen and A. Peterson, "Replication in Computing Education Research: Researcher Attitudes and Experiences," Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16), pp. 2-11, [Online], Available: https://doi.org/10.1145/2999541.2999554, 2016.

[5] N. C. C. Brown, M. Kölling, D. McCall and I. Utting, "Blackbox: A Large-scale Repository of Novice Programmers' Activity," Proc. of the 45th ACM Tech. Symposium on Computer Science Education (SIGCSE '14), pp. 223-228, [Online], Available: https://doi.org/10.1145/2538862.2538924, 2014.

[6] D. Zingaro, Y. Cherenkova, O. Karpova and A. Petersen, "Facilitating Code-writing in PI Classes," Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE ’13), pp. 585-590, [Online], Available: https://doi.org/10.1145/2445196.2445369, 2013.

[7] A. Papancea, J. Spacco and D. Hovemeyer, "An Open Platform for Managing Short Programming Exercises," Proceedings of the 9th Annual Int. ACM Conference on International Computing Education Research (ICER ’13), pp. 74-52, [Online], Available: https://doi.org/10.1145/2493394.2493401, 2013.

[8] A. Vihavainen, T. Vikberg, M. Luukkainen and M. Pärtel, "Scaffolding Students’ Learning Using Test My Code," Proc. of the 18th ACM Conf. on Innovation and Technology in Computer Science Education (ITiCSE ’13), pp. 117-122, [Online], Available: https://doi.org/10.1145/2462476.2462501, 2013.

[9] C. Piech, M. Sahami, D. Koller, S. Cooper and P. Blikstein, "Modeling How Students Learn to Program," Proc. of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE ’12), pp. 153-160, [Online], Available: https://doi.org/10.1145/2157136.2157182, 2012.

[10] V. Karavirta, A. Korhonen and L. Malmi, "On the Use of Resubmissions in Automatic Assessment Systems," Computer Science Education, vol. 16, no. 3, p. 229–240, 2006.

[11] P. Blikstein, "Using Learning Analytics to Assess Students’ Behavior in Open-ended Programming Tasks," Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK ’11), pp. 110-116, [Online], Available: https://doi.org/10.1145/2090116.2090132, 2011.

[12] P. Blikstein, M. Worsley, C. Piech, M. Sahami, S. Cooper and D. Koller, "Programming Pluralism: Using Learning Analytics to Detect Patterns in the Learning of Computer Programming," Journal of Learning Sciences, vol. 23, no. 4, p. 561–599, 2014.

[13] A. Allevato and S. H. Edwards, "Discovering Patterns in Student Activity on Programming Assignments," ASEE Southeastern Section Annual Conference and Meeting, [Online], Available: http://people.cs.vt.edu/~edwards/index2.php?option=com_content&do_pdf=1&id=288, Virginia Polytechnic Institute and State University Blacksburg, Virginia, 2010.

[14] A. Altadmri and N. C. C. Brown, "37 Million Compilations: Investigating Novice Programming Mistakes in Large-scale Student Data," Proceedings of the 46th ACM Technical Symposium on Computer Science Education, pp. 522- 527, [Online], Available: https://doi.org/10.1145/2676723.2677258, 2015.

[15] N. C. C. Brown and A. Altadmri, "Novice Java Programming Mistakes: Large-scale Data *vs.* Educator Beliefs," ACM Transactions on Education (TOCE), vol. 17, no. 2, pp. 1-21, 2017.

[16] E. S. Tabanao, M. M. T. Rodrigo and M. C. Jadud, "Predicting at-Risk Novice Java Programmers through the Analysis of Online Protocols," Proceedings of the 7th International Workshop on Computing Education Research, pp. 85-92, [Online], Available: https://doi.org/10.1145/2016911.2016930, 2011.

[17] M. C. Jadud, "Methods and Tools for Exploring Novice Compilation Behaviour," Proceedings of the 2nd International Workshop on Computing Education Research, pp. 73-84, [Online], Available: https://doi.org/10.1145/1151588.1151600, 2006.

[18] M. C. Jadud and B. Dorn, "Aggregate Compilation Behavior: Findings and Implications from 27,698 Users," Proceedings of the 11th Annual International Conference on International Computing Education Research, pp. 131-139, [Online], Available: https://doi.org/10.1145/2787622.2787718, 2015.

[19] Ma. Mercedes T. Rodrigo, T. C. S. Andallaza, F. E. V. G. Castro, M. L. V. Armenta, T. T. Dy And M. C. Jadud, "An Analysis Of Java Programming Behaviors: Affect, Perceptions And Syntax Errors among Low-achieving, Average And High-achieving Novice Programmers," Journal Of Educational Computing Research, vol. 49, no. 3, Pp. 293-325, 2013.

[20] N. Ayewah, W. Pugh, D. Hovemeyer, J. D. Morgenthaler and J. Penix, "Using Static Analysis to Find Bugs," IEEE Software, vol. 25, pp. 22–29, 2008.

[21] J. Zheng, L. Williams, N. Nagappan, W. Snipes, J. P. Hudepohl and M. A. Vouk, "On the Value of Static Analysis for Fault Detection in Software," IEEE Transactions on Software Engineering, vol. 32, no. 4, pp. 240-253, 2006.

[22] N. Ayewah, W. Pugh, J. D. Morgenthaler, J. Penix and Y. Zhou, "Evaluating Static Analysis Defect Warnings on Production Software," Proceedings of the 7th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering, pp. 1-8, [Online], Available: https://doi.org/10.1145/1251535.1251536, 2007.

[23] J. R. Ruthruff, J. Penix, J. D. Morgenthaler, S. Elbaum and G. Rothermel, "Predicting Accurate and Actionable Static Analysis Warnings: An Experimental Approach," Proceedings of the 30th International Conference on Software Engineering (ICSE‘08), pp. 341-350, [Online], Available: https://doi.org/10.1145/1368088.1368135, 2008.

[24] A. Bessey, K. Block, B. Chelf, A. Chou, B. Fulton, S. Hallem, C. Henri-Gros, A. Kamsky, S. McPeak and D. Engler, "A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World," Communications of the ACM, vol. 53, no. 2, pp. 66-75, 2010.

[25] S. A. Mengel and V. Yerramilli, "A Case Study of the Static Analysis of the Quality of Novice Student Programs," Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '99), pp. 78-82, [Online], Available: https://doi.org/10.1145/299649.299689, 1999.

[26] S. Nutbrown and C. Higgins, "Static Analysis of Programming Exercises: Fairness, Usefulness and a Method for Application," Computer Science Education, vol. 26, pp. 104–128, 2016.

[27] N. Truong, P. Roe and P. Bancroft, "Static Analysis of Students' Java Programs," Proceedings of the 6th Australasian Conference on Computing Education (ACE '04), vol. 30, pp. 317-325, 2004.

[28] D. Liu and A. Petersen, "Static Analyses in Python Programming Courses," Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), pp. 666-671, [Online], Available: https://doi.org/10.1145/3287324.3287503, 2019.

[29] T. Flowers, C. A. Carver and J. Jackson, "Empowering Students and Building Confidence in Novice Programmers through Gauntlet," Proceedings of the 34th Annual Frontiers in Education (FIE 2004), vol. 1, pp. T3H/10-T3H/13, Savannah, GA, 2004.

[30] M. Hristova, A. Misra, M. Rutter and R. Mercuri, "Identifying and Correcting Java Programming Errors for Introductory Computer Science Students," ACM SIGCSE Bulletin, vol. 35, no. 1, pp. 153–156, 2003.

[31] M. Striewe and M. Goedicke, "A Review of Static Analysis Approaches for Programming Exercises," Proc. of the International Computer-assisted Assessment Conference, Research into E-Assessment (CAA 2014), Communications in Computer and Information Science, vol. 439, pp. 100-113, 2014.

[32] K. Abd Rahman and M. J. Nordin, "A Review on the Static Analysis Approach in the Automated Programming Assessment Systems," Proc. of the National Conference on Programming, [Online], Available:http://www.academia.edu/download/30480008/khirulnizam-reviewonstaticanalysisapproach.

pdf, 2007.

[33] S. H. Edwards, N. Kandru and M. Rajagopal, "Investigating Static Analysis Errors in Student Java Programs," Proceedings of the 2017 ACM Conference on International Computing Education Research, pp. 65-73, [Online], Available: https://doi.org/10.1145/3105726.3106182, 2017.

[34] H. Keuning, B. Heeren and J. Jeuring, "Code Quality Issues in Student Programs," Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE ’17), pp. 110-115, [Online], Available: https://doi.org/10.1145/3059009.3059061, 2017.

[35] T. Delev and D. Gjorgjevikj, "Static Analysis of Source Code Written by Novice Programmers," Proceedings of the 2017 IEEE Global Engineering Education Conference (EDUCON), DOI: 10.1109/EDUCON.2017.7942942, Athens, Greece, 2017.

[36] T. Schorsch, "CAP: An Automated Self-assessment Tool to Check Pascal Programs for Syntax, Logic and Style Errors," Proceedings of the 26th SIGCSE technical symposium on Computer science education (SIGCSE '95), pp. 168–172, [Online], Available: https://doi.org/10.1145/199688.199769, March 1995.

[37] B. Hanks, S. Fitzgerald, R. McCauley, L. Murphy and C. Zander, "Pair Programming in Education: A Literature Review," Computer Science Education, vol. 21, pp. 135–173, 2011.

[38] L. Murphy, B. Richards, R. McCauley, B. B. Morrison, S. Westbrook and T. Fossum, "Women Catch up: Gender Differences in Learning Programming Concepts," Proceedings of the 37th SIGCSE technical symposium on Computer science education (SIGCSE '06), pp 17–21, [Online], Available: https://doi.org/10.1145/1121341.1121350, 2006.

[39] W. W. F. Lau and A. H. K. Yuen, "Gender Differences in Learning Styles: Nurturing a Gender and Style Sensitive Computer Science Classroom," Australasian Journal of Educational Technology, vol. 26, no. 7, [Online], available: https://doi.org/10.14742/ajet.1036, 2010.

[40] N. Subrahmaniyan, L. Beckwith, V. Grigoreanu, M. Burnett, S. Wiedenbeck, V. Narayanan, K. Bucht, R. Drummond and X. Fern, "Testing *vs.* Code Inspection *vs.* What Else?: Male and Female End-users' Debugging Strategies," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08), pp. 617-626, [Online], Available: https://doi.org/10.1145/1357054.1357153, 2008.

[41] L. Beckwith, C. Kissinger, M. Burnett, S. Wiedenbeck, J. Lawrance, A. Blackwell and C. Cook, "Tinkering and Gender in End-user Programmers' Debugging," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), pp. 231-240, [Online], Available: https://doi.org/10.1145/1124772.1124808, 2006.

[42] M. Burnett, S. D. Fleming, S. Iqbal, G. Venolia, V. Rajaram, U. Farooq, V. Grigoreanu and M. Czerwinski, "Gender Differences and Programming Environments: Across Programming Populations," Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10), Article no. 28, pp. 1–10, [Online], Available: https://doi.org/10.1145/1852786.1852824, 2010.

[43] I. Wagner, "Gender and Performance in Computer Science," ACM Transactions on Computing Education (TOCE), vol. 16, no. 3, pp. 1-16, 2016.

[44] A. Lishinski, A. Yadav, J. Good and R. Enbody, "Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals and Self-efficacy on Performance," Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16), pp. 211-220, [Online], Available: https://doi.org/10.1145/2960310.2960329, 2016.

[45] M. Fisher and A. Cox, "Gender and Programming Contests: Mitigating Exclusionary Practices," Informatics in Education, vol. 5, no. 1, pp. 47–62, 2006.

[46] J. Carter and T. Jenkins, "Spot the Difference: Are There Gender Differences in Coding Style?," Proceedings of the 3rd Annual LTSN-ICS Conference, [Online], Available: https://kar.kent.ac.uk/id/eprint/13757, 2002.

[47] L. Williams, R. R. Kessler, W. Cunningham and R. Jeffries, "Strengthening the Case for Pair Programming," IEEE Software, vol. 17, no. 4, pp. 19–25, 2000.

[48] C. McDowell, L. Werner, H. E. Bullock and J. Fernald, "The Impact of Pair Programming on Student Performance, Perception and Persistence," Proceedings of the 25th International Conference on Software Engineering (ICSE '03), pp. 602-607, 2003.

[49] N. Salleh, E. Mendes and J. Grundy, "Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review," IEEE Transactions on Software Engineering, vol. 37, no. 4, pp. 509–525, 2011.

[50] T. Dybå, E. Arisholm, D. I. K. Sjøberg, J. E. Hannay and F. Shull, "Are Two Heads Better Than One? On the Effectiveness of Pair Programming," IEEE Software, vol. 24, no. 6, pp. 12–15, 2007.

[51] S. H. Edwards., "Using Software Testing to Move Students from Trial-and-error to Reflection-in-action," Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04), pp. 26-30, [Online], Available: https://doi.org/10.1145/971300.971312, 2004.

[52] A. Hellas, P. Ihantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom and S. N. Liao, "Predicting Academic Performance: A Systematic Literature Review," Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITICSE'18), pp. 175-199, [Online], Available: https://doi.org/10.1145/3293881.3295783, 2018.

[53] H. Seo, C. Sadowski, S. Elbaum, E. Aftandilian and R. Bowdidge, "Programmers' Build Errors: A Case Study (at Google)," Proceedings of the 36th International Conference on Software Engineering (ICSE 2014), pp 724–734, [Online], Available: https://doi.org/10.1145/2568225.2568255, 2014.

[54] J. Leinonen, L. Leppänen, P. Ihantola and A. Hellas, "Comparison of Time Metrics in Programming," Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER'17), pp. 200–208, [Online], Available: https://doi.org/10.1145/3105726.3106181, 2017.

[55] J. Bennedsen and M. E. Caspersen, "Failure Rates in Introductory Programming," ACM SIGCSE Bulletin, vol. 39, no. 2, pp. 32-36, [Online], Available: https://doi.org/10.1145/1272848.1272879, 2007.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

ملخص البحث:

يمكن أن يُساعد تحليل البيانات المتعلقة بالبرامج التي يكتبها الطّلبة الباحثين في فهم الكيفية التي يتعلم بها المبرمجون المبتدئون، وأن يبين للمعلمين كيف يقومون بتعليم المبتدئين على النحو الافضل. يستكشف هذا البحث كيفية استخدام أدوات التحليل الإستاتيكية للبرامج التي يكتبها الطلبة في محاولةٍ للوقوف على سلوك الطلبة البرمجي وأدائهم في مساقات البرمجة.

تم في هذا البحث تحليل استخدام ثلاث أدوات للتحليل الإستاتيكي في واجبات مقرَّرٍ تمهيدي في البرمجة. وأكدت النتائج ما توصلت إليه دراسات سابقة من أنّ المشاكل المرتبطة بالتنسيق والتوثيق هي المشاكل الأكثر شيوعاً من بين تلك التي وجدت في برامج الطلبة من قبل أدوات التحليل الإستاتيكي، وأنّ هذا الأمر ثابت بصرف النظر عن التخصص والأداء في المقرّر، وأنّ هناك أنواعاً معينة من المشاكل أكثر ارتباطاً بالأداء من غيرها.

كذلك تبيّن أن التكرار الكلي للمشاكل التي تكتشفها أدوات التحليل في واجبات الطالب البرمجية يرتبط بعلامة الطالب النهائية في المقرّر، وأنّ وجود أي نوع من المشاكل في آخر محاولة في كل واجب برمجي يرتبط بضعف الأداء في الاختبارات الغير برمجية في المقرر. من جهة أخرى، اتّضح أنّ الطالبات يرتكبن عدداً أقل من أخطاء التوثيق والأسلوب البرمجي مقارنةً بأقرانهنّ من الذكور، وأنّ الطلبة الذين يعملون بالاشتراك مع غيرهم من الطلبة يرتكبون على وجه العموم أخطاءً أقل مقارنةً بالطلبة الذين يعملون منفردين. كذلك فإن نتائج هذا البحث تطرح أسئلة حول جدوى استخدام بعض المقاييس المستخدمة عادة مع أدوات التحليل البرمجي لتقدير صعوبة تصحيح الأخطاء من جانب الطلبة.

DEVOPS PROCESS MODEL ADOPTION IN SAUDI ARABIA: AN EMPIRICAL STUDY

Mohammad Zarour[1], Norah Alhammad[1], Mamdouh Alenezi[1] and Khalid Alsarayrah[2]

(Received: 15-Mar.-2020, Revised: 2-May-2020, Accepted: 18-May-2020)

ABSTRACT

Nowadays, IT organizations are not seeing DevOps as a competitive advantage or added value, but how can organizations survive if not adopting it? Many software development organizations are adopting DevOps software processes to foster better collaboration between development and operation teams, to improve the software development process's quality and efficiency; therefore, it's very important to measure the adoption of DevOps by these organizations. Maturity models are used as a tool to assess the effectiveness of organizational processes on adopting certain practices and identify what capabilities they need to acquire next to improve their performance and reach a higher maturity level. Few DevOps maturity models have, recently, emerged as a means to assess DevOps adopted practices. This research aims to conduct an empirical field study to assess the DevOps adoption level in seven Saudi organizations using one of the published DevOps maturity models; namely, the Bucena model. The findings show that the adoption of DevOps in the surveyed Saudi organizations is promising; despite that, some factors related to DevOps culture, process and technology are weak and need more attention to enhance them to achieve better performance and continuous delivery.

KEYWORDS

DevOps process model, Empirical study, Process adoption, Process assessment, Software process.

1. INTRODUCTION

Software engineers have noticed the gap between development and operation teams. The collaboration between the two teams has been discussed where this discussion yielded a new process model called DevOps. Nowadays, IT organizations are not seeing DevOps as a competitive advantage or added value, but how can organizations survive if not adopting it? The new process model aims to achieve fast high-quality releases of the software product. DevOps is the new software process that extends the agility practices within the collaborative culture to enhance the process of software development and delivery [1]-[2]. Moreover, the DevOps approach is concerned with improving the collaboration between the development and operation teams, which represents a new shift in understanding the way to build software systems. Both the development team and the operation team have different goals in the project; developers' goal is to release the new features of the software to production, whereas the operators' goal is to keep the software as stable and available as possible. To maintain these two goals, the collaboration between development and operation teams as early as possible in the project is vital. This collaboration is not considered in agile methodologies to achieve fast, high-quality software releases.

Hence, DevOps changes the workflow of traditional software development to accelerate and streamline software delivery, which means changing not only the process flow, but also the organizational culture in developing and delivering software. This means that adopting DevOps may spur the organization to introduce new processes, personnel and technological change [3]. Nowadays, software delivery is treated as a continuously evolving process to meet user expectations. DevOps makes this possible by bringing the development and operation teams together to facilitate collaboration, continuous integration, continuous delivery and automation, which will result in reduced time to market, enhanced customer experience, improved quality assurance and reduced costs.

According to Forrester data [4], more than 50% of organizations across industries, including healthcare, manufacturing and banking organizations, have already incorporated DevOps as part of their digital strategy. Although DevOps has gained recently more interest in academia and practice, the literature still has a limited amount of academic publications as well as empirical studies related to DevOps [5]. Consequently, few maturity models and empirical studies related to DevOps exist in the literature. One

1. M. Zarour, N. Alhammad and M. Alenezi are with Department of Software Engineering, Prince Sultan University, Riyadh, Saudi Arabia. Emails: mzarour@psu.edu.sa, norah.a.alhammad@psu.edu.sa and malenezi@psu.edu.sa
2. K. Alsarayrah is with Department of Software Engineering, Hashemite University, Zarqa, Jordan. Email: khalidt@hu.edu.jo

of these maturity models will be used in this study, based on a recommendation of a previous study [6], to conduct a new empirical study to assess the maturity of DevOps adoption in several Saudi organizations. The main purpose of this study is to assess the maturity of DevOps adoption level in Saudi organizations as the DevOps process model is new and those organizations have no idea about how good they are in adopting it. For this research, a survey study has been designed to assess the maturity of DevOps adoption in Saudi organizations. The research conducted in this paper is a multi-case study that collects data *via* interviews and surveys. The research methodology adopted in this research is summarized in steps as follows:

1. Reviewing available maturity models and choosing one to use in the empirical study.
2. Developing the assessment method (interview/survey questions).
3. Analyzing and discussing findings.
4. Reporting findings

The first step of this research is a theoretical one that aims to review available DevOps maturity models, which is briefly discussed in the next section. For more details about this step and detailed comparison, please refer to [6]. Then, an assessment method is developed to be used in evaluating DevOps adoption in various Saudi organizations. The second part of this methodology is concerned conducting the empirical study that assesses the maturity of DevOps adoption in Saudi Arabia, which is discussed in Sections 4 and 5 of this research.

2. LITERATURE REVIEW

Software products can be improved to deliver better results by adopting DevOps practices. The software product quality will increase when adopting DevOps practices that consider the strong relationship between culture, automation, measurement and sharing, as they enhance quality [7]. DevOps is consisting of practices and cultural values to minimize the barriers between development and operation teams and DevOps adoption involves a tight relationship between agility, automation, collaborative culture, continuous measurement, quality assurance, resilience, sharing and transparency [8]. Therefore, using practices that make the software product available and ready to the requester as soon as it gets implemented leads to the importance of understanding how the deployment practices are applied in the development team and that can be achieved through establishing a proper maturity model and using it to survey the development team and operation team and their practices [9]. A qualitative case study for three software development companies in Finland was conducted to indicate the benefits and challenges involved in adopting DevOps in 2016 [10]. Another case study in Finland was conducted to measure the impact of DevOps adoption focusing on mixing responsibilities between the development and operation teams [11]. From the result of that study, DevOps practices are influencing positively software products in terms of released duration and quality. A few years ago, it was claimed that no dedicated maturity models for DevOps exist [12]. Recently, few DevOps maturity models have been developed and documented in the literature. Most of them are based on capability maturity mode as this model and its related models are found feasible to guide the process improvement for DevOps processes [13]. In the coming few paragraphs, we discuss a sample of these models briefly.

First, Bahrs form IBM provided an analysis of the adoption of the IBM DevOps approach for promoting continuous delivery of software [14]. The author identified four dimensions in adopting or implementing continuous software growth within an organization. These dimensions include: planning and measuring, developing and testing, releasing and deploying and monitoring and optimizing. The maturity mode was defined with four levels that are: practiced, consistent, reliable and scaled. The IBM DevOps maturity model is practice-based and reflects a wider context within the adoption framework of a software development organization. It focuses on defining the best practices to be applied in the adoption of new software solutions iteratively. The main strengths of this approach include the fact that it provides a well-articulated way for assessing current DevOps practices within an organization. It also helps in defining a clear roadmap for DevOps implementation.

Second, a DevOps maturity model was proposed by Mohamed and used to assess global software development practices and processes [3]. The proposed DevOps maturity model is based on the Capability Maturity Model Integration (CMMI) and is composed of five maturity levels, which are: initial, managed, defined, measured and optimized against four dimensions that include: quality, automation, collaboration and governance. CMMI helps organizations improve productivity, reduce defects, optimize the process and ensure predictability and efficiency of operations [15]. Adopting the CMMI model to develop the DevOps maturity model will help achieve these benefits in the domain of DevOps processes.

Third, another suggested DevOps maturity model based on (CMMI) process model is proposed in [12]. The model uses a combination of CMMI for development (CMMI-DEV) and CMMI for services (Dev-SVC) to evaluate the maturity of adopting DevOps. That approach was tested on a specific software project at a very large telecom organization, where there were more than 100 developers and 8 operation people working on that project. The software project was adopting DevOps and a test assessment was conducted using SCAMPI C assessment, which is the least formal assessment method. The results showed that the use of CMMI model (CMMI-Dev and CMMI-SVC) can support the purpose of evaluating the processes within software projects that are adopting DevOps practice. However, that study tested only the processes at level 2 (managed) and level 3 (defined).

Fourth, another model that is aligned with the CMMI maturity for DevOps adoption was presented by the employees from Hewlett Packard Enterprise (HPE) [16]. That model is designed to cover the entire lifecycle of an application for large organizations, regardless of the change being determined by the development team or the IT operation teams. It was applied to measure process, automation and collaboration dimensions and the levels of this model are: initial, managed, defined, measured and optimized.

Fifth, a suggested maturity model with five levels is proposed in [17]. The model has three dimensions; namely: people, process and tools. The model levels are: basic, emerging, co-ordinate, enhanced and top level.

Sixth, maturity model for DevOps with four levels has been proposed in [18], known as the Bucena maturity model. Its levels are: initial, repeatable, defined, managed and optimized. The model has four DevOps dimensions, which are:

- Culture: Adopting DevOps requires a new culture that supports transparency and a good supporting environment between the development and operation teams [19]. This can be achieved by arranging regular meetings between both teams. The development team should be supportive of the operation team during the release and production of the software [13].
- People: The team members of a DevOps process should be skilled persons with high ability in improving their skills via self-learning and team-learning. Team members should show a high level of collaboration and support for each other.
- Process: The DevOps process focuses on continuous delivery, continuous testing, continuous integration and continuous monitoring. The DevOps processes adopt agile practices in development.
- Technology: This dimension discusses the technologies and tools support the DevOps process in continuous delivery and to bring the development and operation environments to work collaboratively. It also provides various automation technologies to support the process dimension, hence increase productivity.

Bucena's model [18] revolves around the provision of a DevOps methodology for implementation within small enterprises. Note that the authors of this model promote this model as part of helping very small entities to adopt DevOps via three steps, see [20]. Although this maturity model is focusing on very small entities, it can still be used for anyone interested in adopting it or experiencing its benefits [20]. Accordingly, this research paper adopts the Bucena maturity model to conduct the field study, because it assesses the culture dimension which is a critical dimension to improve software quality in a DevOps environment [7]-[8] and because the model is originated from academia, not white papers. We focus in this paper only on one step, which is assessing the current organizational DevOps maturity level while leaving the issue of how to enhance the DevOps maturity level for each organization to decide on it.

3. Data Collection

The sampling that was chosen to be part of this study consists of seven Saudi organizations that provide software products and have started practicing DevOps. These organizations are notated as organization A, B, C, D, E, F and G and are briefly introduced as follows:

1. Organization A: A Saudi governmental organization that provides various IT serves to the public.
2. Organization B: A Saudi organization that provides IT services to governmental agencies.
3. Organization C: A Saudi organization that provides healthcare information technology (HIT) solutions
4. Organization D: A business services Saudi organization that is focusing on implementing smart solutions, business services and data services.
5. Organization E: A Saudi organization that provides and supports custom and commercial off-the-shelf solutions in different sectors, such as health sectors.
6. Organization F: Saudi organization that supports digital infrastructure, information security and system development.
7. Organization G: A Saudi organization that provides modern information technology and communication manufacturing, system integration, as well as operation and maintenance services in Saudi Arabia.

Two types of data generating methods are used in this study: First, a structured interview that consists of demographic questions with a senior-level employee in each organization to collect demographic data about the organization, like organization size, organization domain, size of the delivery team, the duration of experiencing agile development and the duration of adopting DevOps. Tables 1 summarizes this information for the seven organizations. Second, a questionnaire is developed by using an online survey tool (SurveyMonkey). This questionnaire is designed based on the Bucena DevOps maturity model.

Table 1. Summary of organizations.

Org.	**Domain**	**Size**	**Delivery team size**	**Representative position**	**Agile**	**DevOps**
A	Business services	Large	~ 13	Senior technical manager	1 year	8 months
B	IT solutions	Large	~ 10	Senior manager of software development	2 years	6 months
C	Application service provider (ASP)	Medium	~ 7	Technical team leader	5 years	9 months
D	Business services	Large	~ 13	Senior technical consultant	3 years	1 year
E	Business and IT services	Very Large	~ 9	DevOps transformation leader	5 years	1 year
F	Software service provider (SSP)	Medium	~ 15	Senior software developer	1 year	3 months
G	IT solutions	Very Large	~ 8	Senior software engineer	3 years	1 year

The goal of the questionnaire is to assess the DevOps maturity level of Saudi organizations. The questionnaire consists of 23 questions grouped as follows: 9 questions related to DevOps technology dimension, 6 questions related to the DevOps process dimension, 5 questions related to DevOps culture dimension and 3 questions related to DevOps people dimension. Table 2 illustrates a sample of the mapping of the DevOps dimensions, DevOps factors, questions and possible answers. Moreover, each question represents a single factor in a dimension and a score is assigned for each question/factor: level 1 for the first option, level 2 for the second option, level 3 for the third option, level 4 for the fourth option and level 5 for the fifth and other options. Appendix A provides a list of questions developed for the interview and the questionnaire. Note that the factors for each dimension are identified and discussed in the Bucena maturity model, while the description of each maturity level for each factor is used to devise the survey questions and possible answers. For more information about the maturity model refer to [18].

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

Note that the CMMI maturity model provides two representations; the continuous representation where the processes are assessed individually according to their capability level and the staged representation where the whole organization is assessed according to its maturity level. For more details about CMMI maturity model, refer to [13]. In this paper, where Bucena maturity model is used to assess DevOps maturity and at the time of writing this paper and conducting the case study, no details have been given about this model in terms of how to calculate the capability level or maturity level of the organization. Hence, the authors of this paper have calculated the dimension capability level based on the dimension level formula, see formula number 1. This formula is used to calculate the capability level, because some of the questions are out of four and not all of them are out of five. Moreover, the organization maturity level is calculated by summing up the four dimensions' capability levels and dividing them by four (number of DevOps dimensions).

$$Dimension\ Capability\ Level = \frac{\sum_{i=1}^{n} f_i\ Gained\ Score}{\sum_{i=1}^{n} f_i\ Maximum\ Score}\ X\ l \qquad (1)$$

where: f = the factor in the desired dimension, N = number of factors in the desired dimension l = 5 (number of levels).

Table 2. Sample of DevOps dimensions, questions and possible answers.

Technology Dimension		
Factor	Assessment question	Possible answer
Required Environment	How are the required environments provisioned?	1. Environments are provisioned manually.
		2. All environment configurations are externalized and versioned.
		3. Virtualization used if applicable.
		4. All environments are managed effectively.
		5. Provisioning is fully automated.
Deliverables' Validation	What is the level of validation of the deliveries developed?	1. Manual tests or minimal automation.
		2. Functional test automation.
		3. Triggered automated tests.
		4. Smoked tests and dashboard shared with operation team.
		5. Chaos Monkey or other tools are used to test resilient to instance failures.
Deployment Automation	What is the level of deployment automation in your organization?	1. Manual deployment.
		2. Build automation.
		3. Non-production deployment automation.
		4. Production deployment automation.
		5. Operation and development teams regularly collaborate to manage risks and reduce cycle time.

4. Results and Analysis

After conducting the questionnaire on the seven organizations from different industries in Saudi Arabia, the maturity level was measured (level 1: initial, level 2: repeatable, level 3: defined, level 4: managed and level 5: optimized) against the factors associated with each dimension (technology, process, people and culture). Note that the dimensions rated less than 3 out of 5 are considered weakness points and need more attention to improve them and their factors.

4.1 Technology Dimension

The technology dimension has nine factors to assess its capability. Five of these factors are having the possible maximum value of 5 and the other ones are having the possible maximum value of 4. Table 3 shows the achieved level for each factor in the technology dimension among the surveyed organizations. The analysis of the technology factors shows that six out of the nine factors are showing good average levels; i.e., achieved average level 3 or above, among the surveyed organizations.

Although all organizations achieved good capability levels for the technology dimension; i.e., achieved level 3 or above, three technology factors achieved low average levels, e.g. less than 3; namely, data management, software configuration management and issue tracking. These technology factors need more focus by all organizations. In other words, tools and automation of these factors should be activated and properly adopted by organizations.

Table 3. Technology dimension – factors' levels (out of 5).

Technology's factors	Max.	Org. A	Org. B	Org. C	Org. D	Org. E	Org. F	Org. G	AVG.	STD.
Required environment	4	3	4	4	4	3	4	3	3.6	0.49
Deliverables' validation	4	1	4	3	4	2	3	4	3.0	1.07
Data management	4	2	4	3	3	1	3	3	**2.7**	0.88
Deployment automation	5	5	4	5	5	2	2	4	3.9	1.25
Build management	5	2	5	4	4	2	2	2	3.0	1.20
Collaboration	4	3	4	3	4	4	3	3	3.4	0.49
Software configuration management	4	3	3	4	2	2	3	3	**2.9**	0.64
Data monitoring	4	3	4	3	4	4	1	2	3.0	1.07
Issue tracking	2	1	2	2	2	1	2	2	**1.7**	0.45
Technology capability level	5	3	5	4	4	3	3	4	3	1

4.2 Process Dimension

The process dimension has six factors to measure the process's maturity. These factors are delivery, development, testing, project management, documentation and organization processes, see Table 4.

The analysis of the process factors shows that two out of the six factors are showing good average levels; i.e., achieved average level 3 or above, among the surveyed organizations. The delivery and testing process factors are considered the best factors in the process dimension. Although all organizations achieved good capability levels for the process dimension; i.e., achieved level 3 or above, four process factors achieved low average levels, e.g. less than 3; namely, the development process, the project management process, the documentation process and the organization process. These process factors need more focus from the organizations.

Table 4. Process dimension – factors' levels.

Process's factors	Max. Level	Org. A	Org. B	Org. C	Org. D	Org. E	Org. F	Org. G	AVG	STD.
Delivery process	5	4	5	5	5	2	5	2	4.0	1.31
Development process	3	2	2	3	2	3	2	3	**2.4**	0.49
Testing process	5	2	5	5	5	3	5	2	3.9	1.36
Project management process	4	2	2	3	4	3	3	2	**2.7**	0.70
Documentation process	4	2	3	4	4	1	1	3	**2.6**	1.18
Organization process	5	2	3	5	3	2	2	3	**2.9**	0.99
Process capability level	5	3	4	5	4	3	3	3	3.1	1

4.3 People Dimension

This dimension has three factors to measure its capability, which are team organization, learning process and development of competencies and capabilities. Table 5 shows the achieved level for each factor in people dimension among the surveyed organizations.

The analysis of the people factors shows that two out of the three factors are showing good average levels; i.e., achieved average level 3 or above, among the surveyed organizations. Two factors achieved low average levels, e.g. less than 3; namely, team organization and development of competencies and capabilities.

Table 5. People dimension – factors' levels.

People's factors	Max. Level	Org. A	Org. B	Org. C	Org. D	Org. E	Org. F	Org. G	AVG.	STD.
Team organization	5	5	4	5	4	4	1	4	3.9	1.25
Learning process	5	2	2	3	5	2	3	2	**2.7**	1.03
Development of competencies and capabilities	5	2	4	5	5	2	5	2	3.6	1.40
People Capability Level	5	3	3	4	5	3	3	3	3.4	1

4.4 Culture Dimension

This dimension has six factors to measure the culture capability, which are: communication type, requirement understanding, culture understanding, collaboration and innovation drivers. Table 6 shows the achieved level for each factor in the culture dimension among the surveyed organizations.

Table 6. Culture dimension – factors' levels.

Culture's factors	Max. Level	Org. A	Org. B	Org. C	Org. D	Org. E	Org. F	Org. G	AVG.	STD.
Communication type	5	4	4	5	4	3	4	3	3.9	0.64
Requirement understanding	5	1	3	5	5	4	5	3	3.7	1.39
Culture understanding	4	3	2	4	4	2	2	1	**2.6**	1.05
Collaboration	4	2	3	4	3	3	3	3	3.0	0.53
Innovation drivers	4	2	2	4	4	2	2	2	**2.6**	0.90
Culture capability level	5	3	3	5	5	3	4	3	3.1	1

The analysis of the cultural factors shows that three out of the five factors are showing good average levels; i.e., achieved average level 3 or above, among the surveyed organizations. Two factors achieved low average levels, e.g. less than 3; namely, cultural understanding and innovation drivers.

Figure 1 illustrates the achieved capability level for each dimension in each of the surveyed organizations.

Table 7 illustrates the calculated maturity levels for each organization, while Figure 2 visually illustrates the maturity levels. Even though the organizations achieved maturity levels 3 or above, Tables 3, 4, 5 and 6 depict that some factors in the different dimensions gained capability levels less than 3 and need improvement. Accordingly, organizations need to pay attention not only to their overall maturity level, but also to the individual capability of each dimension and the level achieved per factor as well. For instance, Organization A has achieved maturity level 3 and its capability level for each dimension is 3, but some factors gained a level less than 3, so these factors need to be improved for better performance of the whole DevOps process.

- Technology: Databases are a key component for the operation team, which means that it is part of the DevOps whole process. Organizations should pay attention to how they manage their databases in an agile manner; i.e., data should be flexible to change quickly and reliably. For instance, organizations should decide how to shift toward a more agile database; whether to keep using traditional relational DBMSs or shift toward other forms, such as NoSQL or Casandra. Another issue is related to adopting more automation in software configuration management as well as issue tracking for changes that may happen during continuous integration and continuous deployment and decide which tools suit them more.

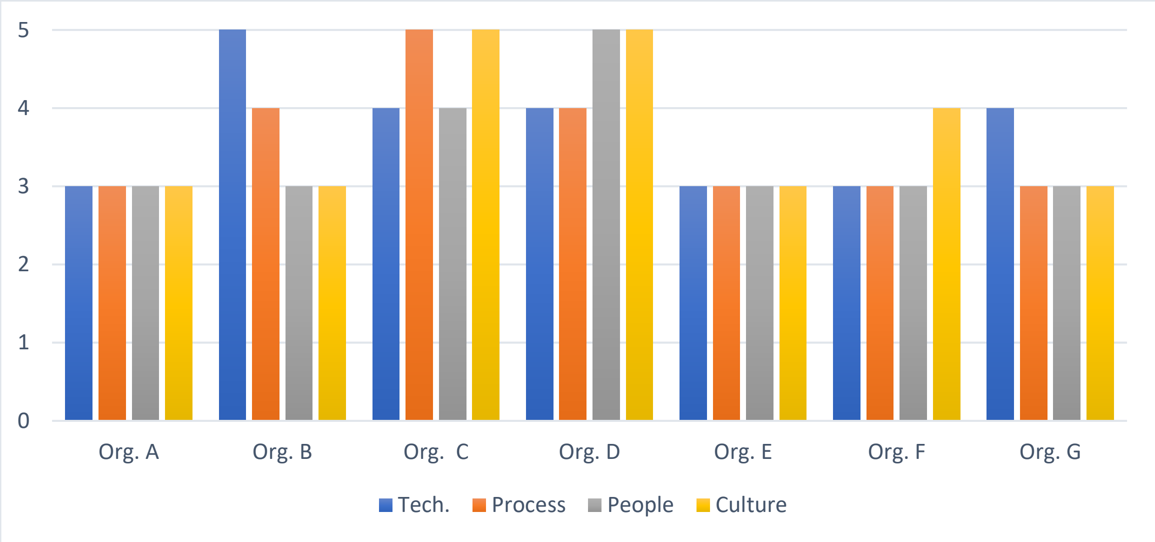


Figure 1. Capability level for each dimension for each organization.

Table 7. Organizations DevOps maturity.

Dimension	Org. A	Org. B	Org. C	Org. D	Org. E	Org. F	Org. G
Tech.	3	5	4	4	3	3	4
Process	3	4	5	4	2	3	3
People	3	3	4	5	3	3	3
Culture	3	3	5	5	3	4	3
Maturity level	3	3	4	4	3	3	3

- Process: Organizations should improve their agile processes and make sure that agile principles and practices are implemented in the organization. Sometimes, organizations claim that they adopt agile development processes, but their processes are run traditionally. Moreover, project managers should think of how they can manage agile processes that support continuous integration and deployment, e.g. project managers may think of adopting xProcess agile planning tool. Documentation in agile processes with short iterations and releases is vital. Organizations need to specify how they document their development process when using DevOps. Organizations can make use of different documentation tools available to automate this process and achieve continuous documentation as well.
- People: Learning is seen as a process rather than an event organized once or twice. Organizations need to adopt a well-organized and planned learning process for their people and engineers to enhance their understanding of the DevOps process.
- Culture: Changing the organizational culture and engineers' mindset to adopt agility and DevOps practices is very critical for success. Engineers need to understand the culture of DevOps, where the development team is not responsible for development only and the operation team is not responsible for releasing and deploying it only, but instead, both teams are responsible for running the product and bring the system alive.

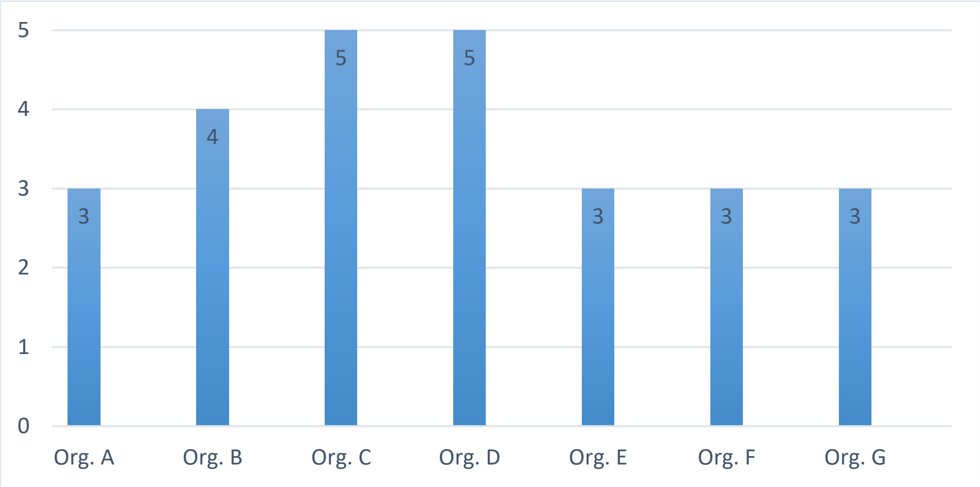


Figure 2. Maturity level per organization.

Table 8. Factors that gained low average ratings for all organizations.

Dimension	Factor	Average (out of 5)
Technology	Data management	2.7
	Software configuration management	2.9
	Issue tracking	1.7
Process	Development process	2.4
	Project management process	2.7
	Documentation process	2.6
	Organization process	2.9
People	Learning process	2.7
Culture	Culture understanding	2.6
	Innovation drivers	2.6

5. VALIDITY THREATS

This research was planned to take into account validity concerns that consist of three categories: construct validity, external validity and reliability [21].

Due to the novelty of the DevOps process, scientific papers that document the DevOps maturity models are few. The DevOps maturity model used in this research, as well as other related maturity models, are published in scientific papers, which in turn, have conducted a literature review to identify these models as well.

Regarding the conducted empirical study using interviews and survey, one of the possible threats that relate to reliability and construct validity is finding a sufficient number of Saudi organizations of different organization sizes that adopt DevOps to assure the reliability of the assessment survey. To solve this issue, we went to GITEX 2018 and visited some of the Saudi governments' booths and asked them about their preferred software provider organizations to consider them in this study.

Also, we met big Saudi companies that participated in GITEX 2018 and asked them whether or not they adopt DevOps. After that, we conducted this study with three government organizations, three semi-government organizations and one private company, with different sizes.

Moreover, another possible threat which is related to construct validity is concerned with representatives' interpretation of some questions. To avoid this, we asked the interviewees to answer the online survey during the face-to-face meeting and after the interview immediately. The questions have been explained to the organizations' representatives before they started filling the survey.

Regarding reliability, the survey is based on the Bucena DevOps maturity model, which is documented in the literature. The model has been validated by the authors in one small enterprise. Furthermore, three independent software engineers have answered the survey as a pilot study, before publishing it to verify its clearness.

Finally, another possible external validity threat is that participants may feel the need to present their company in the best light during the interview and survey. Therefore, to avoid this, it has been communicated to the participants that the results will be published anonymously and answering the questions in the most realistic view would give a better evaluation for their company which will help them identify the strengths and weaknesses of their DevOps adoption.

6. CONCLUSION

In this research, the Bucena DevOps maturity model was chosen to conduct an empirical study on Saudi organizations. Choosing the Bucena model does not mean that other maturity models are useless. On the contrary, other maturity modes deserve practicing to enrich the empirical studies on DevOps maturity models. The empirical study was conducted on seven Saudi organizations by interviewing these organizations' representatives and distributing surveys among them. These organizations vary in size; two organizations are medium-sized, three organizations are large-sized and two organizations are very large-sized.

This research shows a promising future for Saudi organizations in adopting DevOps. Despite this, Saudi organizations need to pay attention to various factors in the different DevOps dimensions which are found to be weakness points. Most importantly, organizations adopting DevOps need to adapt their engineers' mindset to understand and implement DevOps processes properly and enhance their learning process in this regard. Although automation and tools used are in a good position in some phases on the DevOps projects, such as testing, other phases need more work to automate them. This includes configuration management and issue tracking during the continuous delivery process. Moreover, organizations need to think about how they will manage their data and DBMS to meet agile principles and continuous delivery.

One final note for the maturity model developers is that it insufficient to develop the maturity model and its rating scale. IT is better if the assessment method that implements the model is also published along with the model. For instance, the CMMI model has its assessment method, which is known as the SCAMPI method. If the assessment method is not published or documented, researchers will develop their assessment methods to use the maturity model, as we did here and this can result in divergence among these methods.

REFERENCES

[1] L. Leite, C. Rocha, F. Kon, D. Milojicic and P. Meirelles, "A Survey of DevOps Concepts and Challenges," ACM Comput. Surv., vol. 52, no. 6, pp. 1–35, Nov. 2019.

[2] A. Dyck, R. Penners and H. Lichter, "Towards Definitions for Release Engineering and DevOps," Proc. of the IEEE/ACM 3rd International Workshop on Release Engineering (RELENG '15), DOI 10.1109/RELENG.2015.10, 2015.

[3] S. Mohamed, "DevOps Shifting Software Engineering Strategy-value Based Perspective," Int. J. Comput. Eng., vol. 17, no. 2, pp. 51–57, 2015.

[4] Robert Stroud, "2018: The Year of Enterprise DevOps," Forrester, [Online], Available: https://go.forrester.com/blogs/2018-the-year-of-enterprise-devops/, [Accessed on: 08-Feb-2019].

[5] J. Smeds, K. Nybom and I. Porres, "DevOps: A Definition and Perceived Adoption Impediments," Proc.

of the International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2015), vol. 212, pp. 166–177, 2015.

[6] M. Zarour, N. Alhammad, M. Alenezi and K. Alsarayrah, "A Research on DevOps Maturity Models," Int. J. Recent Technol. Eng., vol. 8, pp. 4854–4862, 2019.

[7] P. Perera, M. Bandara and I. Perera, "Evaluating the Impact of DevOps Practice in Sri Lankan Software Development Organizations," Proc. of the 16th International Conference on Advances in ICT for Emerging Regions (ICTer), DOI: 10.1109/ICTER.2016.7829932, pp. 281–287, 2017.

[8] W. P. Luz, G. Pinto and R. Bonifácio, "Building a Collaborative Culture: A Grounded Theory of Well Succeeded DevOps Adoption in Practice," Proc. of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Article no. 6, pp. 1–10, [Online], Available: https://doi.org/10.1145/3239235.3240299, pp. 1–10, 2018.

[9] A. Virtanen, K. Kuusinen, M. Leppänen, A. Luoto, T. Kilamo and T. Mikkonen, "On Continuous Deployment Maturity in Customer Projects," Proc. of the Symposium on Applied Computing (SAC '17), pp. 1205–1212, [Online], available: https://doi.org/10.1145/3019612.3019777, 2017.

[10] S. Riungu-Kalliosaari, L., Mäkinen, L. E. Lwakatare, J. Tiihonen and T. Männistö, "DevOps Adoption Benefits and Challenges in Practice: A Case Study," Proc. of the 17th International Conference on Product-Focused Software Process Improvement (PROFES 2016), pp. 590–597, Trondheim, Norway, 2016.

[11] K. Nybom, J. Smeds and I. Porres, "On the Impact of Mixing Responsibilities between Devs and Ops," Proc. of the International Conference on Agile Software Development, DOI: 10.1007/978-3-319-33515-5_11, pp. 131–143, 2016.

[12] G. Rong, H. Zhang and D. Shao, "CMMI-guided Process Improvement for DevOps Projects: An Exploratory Case Study," Proc. of the International Conference on Software and Systems Process, pp. 76–85, pp. 76-85, [Online], available: https://doi.org/10.1145/2904354.2904372, 2016.

[13] E. Shamow, "DevOps at Advance Internet: How We Got in the Door | Cutter Consortium," Cutter IT Journal, vol. 24, no. 8, pp. 13–18, 2011.

[14] P. Bahrs, "Adopting the IBM DevOps Approach for Continuous Software Delivery," [Online], available: https://developer.ibm.com/technologies/devops/articles/d-adoption-paths/, 2013.

[15] S. Sreenivasan and K. Kothandaraman, "Improving Processes by Aligning Capability Maturity Model Integration and the Scaled Agile Framework," Glob. Bus. Organ. Excell., vol. 38, no. 6, pp. 42–51, Sep. 2019.

[16] S. Inbar et al., "DevOps and OpsDev: How Maturity Model Works-Hewlett Packard," Hewlett Packard Enterprise Community, 2013.

[17] G. Menzel and A. Macaulay, "DevOps - The Future of Application Lifecycle Automation," Capgemini, [Online], Available: https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2016/03/devops-the-future-of-application-lifecycle-automation.pdf, 2015.

[18] I. Bucena and M. Kirikova, "Simplifying the DevOps Adoption Process, " Proc. of BIR Workshops, vol. 1898, Paper no. 14, [Online], Available: http://ceur-ws.org/Vol-1898/paper14.pdf, 2017.

[19] L. E. Lwakatare, P. Kuvaja and M. Oivo, "An Exploratory Study of DevOps Extending the Dimensions of DevOps with Practices," Proc. of the 11th International Conference on Software Engineering Advances (ICSEA), pp. 91–99, Rome, Italy, 2016.

[20] I. Būcena, "DevOps Adoption Method for VSE," [Online], Available: https://devopsadoptmeth.wordpress.com/.

[21] P. Runeson and M. Höst, "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," Empirical Software Engineering, vol. 14, no. 2, pp. 131–164, Apr. 2009.

APPENDIX

Interview questions	
[1] In which industry/ domain is your organization?	[2] What is the size of your organization? Or how many employees are at your company?
[3] Does your company practice agile? If yes, for how many years?	[4] Does your company adopt DevOps? If yes, for how long?
[5] What is your current position at your company?	[6] What is the average size of the delivery team?
[7] Does your team have a different level of understanding DevOps? Why?	[8] Do different members of your team have the same motivation to apply DevOps practices? Why?

DevOps Maturity Assessment: Questions related to the technology key area	
1. How are the required environments provisioned?	
o Environments are provisioned manually.	o All environment configurations are externalized and versioned
o . Virtualization used if applicable.	o All environments are managed effectively.

- Provisioning is fully automated.

2. What is the level of validation of the deliveries developed?
- Manual tests or minimal automation.
- Functional test automation.
- Triggered automated tests.
- Smoked tests and dashboard shared with operational team.
- Chaos Monkey.

3. How is the data management process organized?
- Data migration is un-versioned and performed manually.
- Changes to DB done with automated scripts versioned with application.
- DB changes performed automatically as part of the deployment process.
- DB upgrades and rollbacks are tested with every deployment.
- Feedback from DB performance after each release.

4. What is the level of deployment automation in your organization?
- Manual deployment.
- Build automation.
- Non-production deployment automation.
- Production deployment automation.
- Operation and development teams regularly collaborate to manage risks and reduce cycle time.

5. How is the build management performed?
- Manual processes for building software/ no artifact versioning.
- Regular automated build and testing any builds can be recreated from source.
- An automated build and test cycle every time a change is committed.
- Build metrics gathered, made visible and taken into account.
- Continuous work on process improvement, better visibility and faster feedback.

6. What is the level of collaboration/knowledge flow between team members?
- No collaboration tools.
- Project planning tool.
- Team collaboration/toolset integration.
- Knowledge management tool.
- Others, using more than the previous; please specify.

7. Does your organization use software configuration management (SCM)?
- No SCM.
- Standardized SCM.
- Configuration is delivered with the code.
- Self-healing tools.

8. What is the level of process and data monitoring?
- No or minimal monitoring.
- Core (basic) monitoring.
- Integrated monitoring.
- Analytics/ Intelligence.

9. In what way issues/bugs are tracked?
- No tools or minimal tool usage for issue tracking.
- All issues and bug reports are tracked.
- Issue reporting automation and monitoring.
- Activities based on the received feedback and data.

Questions related to the process key area

10. How is the deployment of new deliveries to production organized?
- Inconsistent delivery process.
- Scheduled delivery process.
- Automated delivery process.
- Frequent delivery process.
- Continuous delivery process.

11. What is the way (type/approach) of the development process in your organization?
- *Ad-hoc* development.
- Scrum development.
- Agile development.
- Lean development.
- Development process integrated with Six sigma.
- Others, using more than the previous, please specify.

12. How is the software testing process organized?
- *Ad-hoc* testing.
- Requirement-based testing.
- Integrated testing.
- Qualitative testing.
- Continuous testing.
- Others, using more than the previous; please specify.

13. How is the project management process organized?
- Inconsistent project management.
- Project & requirement management.
- Integrated project management.
- Quantitative project management.
- Organized performance management.

14. What is the status of the development documentation?
- Deployment and development documentation is not available or is out-of-date.
- Development documentation and relevant configuration files are up-to-date.
- Regular validation of the documentation and related configuration descriptions is provided.
- Documentation process and structure updates based on gathered experience and quality requirements.
- Others; please specify.

15. How are processes managed in your organization?
- Uncontrolled or reactive processes (management is not applied).
- Processes are managed, but are not standardized.
- Processes are standardized across organization.
- Visibility & predictability of the entire process & performance.
- Highly optimized & integrated processes

Questions related to the people key area

16. How are teams organized in your organization?
- Around skillsets.
- Around deliveries.
- Around projects.
- Around products/ business lines.
- Interdisciplinary teams, organized around KPIs.

17. How is the learning process organized in your organization?
- *Ad-hoc* learning.
- Team learning.
- Value stream learning.
- X-process learning.
- External learning.

18. How are competencies & capabilities developed?
- *Ad hoc* approach of competences development.
- Through training and development.
- Analyses and development of competences.
- Use of mentors.
- Continuous capability improvement.

Questions related to the culture key area

19. What is the main type of communication in your organization?
- Restricted communication.
- Rapid intra-team (inside) communication.
- Rapid communication between teams (inter-team).
- Frequent, collaborative communication.
- Rapid feedback.

20. What is the level of requirement understanding?
- Uncommunicated vision.
- Clear delivery requirements.
- Clear project requirements.
- Clear product/business line requirements.
- Clear organization requirements.

21. What are the understanding and usage of culture in the organization?
- Lack of awareness as to how culture is impacting day-to-day business.
- Awareness of cultural aspects that may help or hinder.
- Cultural traits that support business strategies have been identified.
- Culture viewed as an asset to be managed.
- Desired elements of the culture are identified, ingrained and sustainably creating "the way we work here".

22. What is the level of communication and collaboration?
- Poor, *ad-hoc* communication and coordination.
- Managed communication.
- Active collaboration.
- Collaboration based on process measurement, which allows to identify bottlenecks and inefficiencies.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

23. What drives innovations?
- Sub-innovating/no innovations.
- Innovation by design.
- Innovations by necessity.
- Strategic innovation.

ملخص البحث:

في هذه الأيام، لا تنظر شركات تكنولوجيا المعلومات الى نموذج العمليات (ديف أوبس: التطوير / العمليات) على أنه ميزة تنافسية أو قيمة مضافة، وإنما كيف يمكن للشركات أن تحافظ على بقائها لو لم تستخدم هذا النموذج من العمليات. تتبنى العديد من شركات تطوير البرمجيات عمليات برمجة (ديف أوبس) لتحقيق تعاون أكبر بين فريق التطوير وفريق العمليات لتحسين جودة عملية تطوير البرمجيات وفاعليتها. لذا فإن من المهم جداً قياس تبني نموذج (ديف أوبس) من جانب تلك الشركات.

تُستخدم نماذج النضوج كأداةً لتقييم فعالية العمليات في الشركة والمتعلقة بتبني ممارسات معينة وتحديد القدرات التي يتعين امتلاكها من أجل تحسين الأداء لتلك الممارسات والوصول الى مستوى نضوج أعلى. قد ظهر، مؤخرا، عدد قليل من نماذج النضوج المتعلقة بنموذج العمليات (ديف أوبس: التطوير / العمليات) كوسيلة لتقييم الممارسات المتعلقة بهذا النموذج.

يهدف هذا البحث الى إجراء دراسة ميدانية لتقييم مستوى تبني نماذج (ديف أوبس DevOps) من جانب سبع شركات تقنية سعودية تتبنى هذا النموذج باستخدام أحد نماذج النضوج المنشورة، وهو نموذج بوسينا. وبينت النتائج أن مستوى التبني لنموذج (ديف أوبس) في الشركات السعودية المشاركة في هذه الدراسة هو مستوى واحد. وعلى الرغم من ذلك، هناك عدد من العوامل المتعلقة بثقافة (ديف أوبس) وعملياتها وتقنيتها ما زالت ضعيفة وتحتاج الى المزيد من الاهتمام والتحسين لتحقيق أداء أفضل وتسليمٍ مستمر.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

SENTIMENT ANALYSIS AND CLASSIFICATION OF ARAB JORDANIAN FACEBOOK COMMENTS FOR JORDANIAN TELECOM COMPANIES USING LEXICON-BASED APPROACH AND MACHINE LEARNING

Khalid M.O. Nahar[1], Amerah Jaradat[1], Mohammed Salem Atoum[2] and Firas Ibrahim[3]

(Received: 7-Apr.-2020, Revised: 6-Jun.-2020, Accepted: 8-Jun.-2020)

ABSTRACT

Sentiment Analysis (SA) is a technique used for identifying the polarity (positive, negative) of a given text, using Natural Language Processing (NLP) techniques. Facebook is an example of a social media platform that is widely used among people living in Jordan to express their opinions regarding public and special focus areas. In this paper, we implemented the lexicon-based approach for identifying the polarity of the provided Facebook comments. The data samples are from local Jordanian people commenting on a public issue related to the services provided by the main telecommunication companies in Jordan (Zain, Orange and Umniah). The produced results regarding the evaluated Arabic sentiment lexicon were promising. By applying the user-defined lexicon based on the common Facebook posts and comments used by Jordanians, it scored (60%) positive and (40%) negative. The general lexicon accuracy was (98%). The lexicon was used to label a set of unlabeled Facebook comments to formulate a big dataset. Using supervised Machine Learning (ML) algorithms that are usually used in polarity classification, the researchers introduced them to our formulated dataset. The results of the classification were 97.8, 96.8 and 95.6% for Support Vector Machine (SVM), K-Nearest Neighbour (K-NN) and Naïve Bayes (NB) classifiers, respectively.

KEYWORDS

Jordan Telecom, Sentiment analysis, Lexicon-based, Polarity, Facebook comments, Machine learning, NLP.

1. INTRODUCTION

Language Processing (LP) is the field of computer science and artificial intelligence that mainly studies human-computer language interaction [1]. SA and opinion mining is a field of NLP that investigates and analyzes people's opinions, sentiments, evaluations, attitudes and emotions from written language. It is one of the most active research areas in NLP and is also widely studied in data mining, web mining and text mining [2][24].

The important part of information-gathering behaviour has always been to find out what other people think. With the growing availability and popularity of opinion-rich resources, such as online reviews and personal blogs, new opportunities and challenges arise, as people now can actively use information technologies to seek out and understand the opinions of others. Polarity classification can be applied in individual reviews to evaluate the goodness of a certain product [22][25]. The sudden eruption of activity in the area of opinion mining and SA, which deals with the computational treatment of opinion, sentiment and subjectivity in text, has thus occurred at least in part as a direct response to the surge of interest in new systems that deal directly with opinions as a first-class object [3].

To determine whether a sentence, text or any comment expresses a positive or negative sentiment, three main approaches are commonly used: the lexicon-based approach, machine learning approach and a hybrid approach. Figure 1 explains these approaches [4][29]. In this work, we implemented the lexicon-based approach. The reason behind choosing the lexicon-based approach is that both machine learning and hybrid approaches demand a labeled dataset for supervised learning. Also, Jordanians as

1. K. M. O. Nahar and A. Jaradat are with Computer Sciences Department, Yarmouk University, Irbid, Jordan. Emails: khalids@yu.edu.jo and ameera@yu.edu.jo
2. M. S. Atoum is with Computer Science Department, Irbid National University, Irbid, Jordan. Email: cs@inu.edu.jo
3. F. Ibrahim is with Information Systems and Networks Department, The World Islamic Sciences and Education University, Amman, Jordan. Email: firas.alzobi@wise.edu.jo

other Arabs use their dialects and modern Arabized words, letters, symbols, paronomasias and insinuations for expressing their opinions.

Companies (Zain, Orange and Umniah) interconnect through video, voice and data (mainly internet browsing and social media). The cost of communications provided by those companies is too low compared to neighbouring countries and the level of services provided is also very good, but Jordanians do express their opinions, feelings and sentiments about those companies regarding cost, coverage, offers, internet speed, …etc. These types of opinions may be an indicator of continuing or leaving one company to another or from offer to offer. Most of those opinions, feelings and sentiments are expressed using Jordanian different Arabic dialects in addition to lack of using Original Standard Arabic.

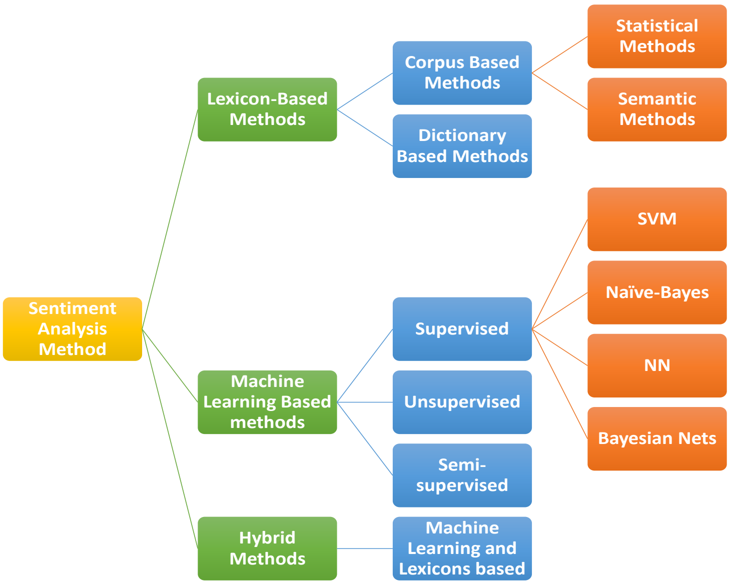


Figure 1. Main approaches of SA [4].

Customer churn analysis is a very common task in data analysis. It involves trying to predict whether customers will quit or continue the contract. It is crucial to the telecommunication companies to review and analyze their customers' feedback to enhance their provided services and avoid losing their contracts. NLP is a great method to automatically analyze sentiments and predict whether those sentiments are positive or negative as an early indicator for the quality of the provided services.

In this work, we are proposing an approach to predict customer satisfaction with the services provided by the telecommunication companies. The approach collects posts and comments from Facebook pages related to Jordanian telecommunication companies in order to find out the customer attitude toward these companies. After collecting and pre-processing the data, sentiment analysis is achieved using the Lexicons-Based Approach (LBA). Owing to the amount of data handled, the work involves automatic translation of English sentiment lexicon to create Arabic sentiment lexicons.

The paper is prearranged as follows. In Section 2, we review some of the previous research related to the field of SA. Then, in Section 3, we introduce the lexicon-based approach for creating the dataset. In Section 4, we apply supervised learning algorithms on the formulated dataset. In addition, we describe the supervised learning model on both KNIME and ORANGE software and show the experimental results and the evaluation of the anticipated method. Finally, we address the conclusions and discuss future works in Sections 5 and 6.

2. Literature Review

Many kinds of research have been devoted to the field of SA [26]-[28]. Unfortunately, few works consider discussing SA for the Arabic language. Moreover, research on Arabic sentiment analysis has not perceived noteworthy developments yet, typically due to the shortage of sentiment resources in Arabic [20]-[21].

Rehab M. Duwairi and Islam Qarqaz [5] carried out an experiment using Rapid miner, which is an open-source machine learning software, to perform SA in Arabic text. The dataset was collected from tweets and Facebook comments that address issues in education, sports and politics. In this study, the main issue was determining the polarity (positive, negative or neutral) of the given text. The authors applied two approaches: the machine learning approach and the lexicon-based approach. Three supervised classifiers (SVM, Naïve Bayes and K-NN) were applied on an in-house collected dataset of 2591 tweets/comments from social media to analyze the sentiment of Arabic reviews. Unfortunately, the dataset was not large enough to make strong conclusions.

Rehab M. Duwairi [6] used classification for SA. After extracting Arabic tweets, the author applied Naïve Bayes (NB) and Support Vector Machine (SVM) classifiers. SVM and NB classifiers were used on a big dataset that consists of almost 22500 Arabic tweets. The experiments involved comparing the lexicon values without the dialect lexicon to the values with converting dialectical words into MSA words. The results show the great impact of the dialect lexicon on the F-measure of the positive and negative classes as well as the Macro-Precision, Macro-Recall and F-Measures. The results were limited by the storage deficiency of the Rapid miner software used.

Ahmad A. Al Sallab et al. [7] concentrated on a deep learning framework to analyze the sentiment of Arabic text with features based on the developed Arabic sentiment lexicon with standard lexicon features. One supervised classifier (SVM) and four unsupervised classifiers (DNN, DBN, DAE and RAE) were applied on a dataset of 3795 entries. Results show that RAE produces the best accuracy.

Haifa K. Aldayel and Aqil M. Azmi [8] proposed a hybrid approach combining semantic orientation and SVM classifiers. The used data passed through pre-processing operations to be ready to a lexical-based classifier, then the output data became a training data for the machine learning classifiers. The proposed approach used 1103 tweets. The experimental results show better F-measure and accuracy of the hybrid approach.

Hala Mulki et al. [9] proposed two classification models to analyze the Arabic sentiment of 3355 tweets written with MSA and Arabic dialects. Authors considered the sentiment classification of Arabic tweets through two classification models: supervised learning-based model and unsupervised learning-based (lexicon-based) model. The conducted experiments showed better F-score and Recall values using the supervised learning-based model. On the other hand, the unsupervised learning-based (lexicon-based) model achieved better results if the stemming did not assign the lookup process.

Nora Al-Twairesh et al. [10] collected a corpus of Arabic tweets by collecting over 2.2 million tweets. Authors presented the sequence of operations used in collecting and constructing a dataset of Arabic tweets: cleaning and pre-processing the collected dataset included filtering, normalization and tokenization. Later, with the help of annotators, the dataset was labeled with (positive, negative, mixed, neutral or indeterminate). Then, the data was classified using the SVM classifier and provided as a benchmark for future work on SA of Arabic tweets.

Hassan Najadat et al. [11] applied four supervised classifiers on a dataset of 4227 posts' texts from the Facebook pages to determine the efficiency of the main three telecommunication companies in Jordan: Orange, Zain and Umniah, based on the SA of customers who use social media, especially Facebook. The results were promising. However, the accuracy without sampling was better than that with sampling.

Leena Lulu and Ashraf Elnagar [12] proposed neural network models from different deep learning classifiers for the automatic classification of Arabic dialectical text. The proposed approach used the manually annotated Arabic online commentary (AOC) dataset that consists of 110 K labeled sentences. This approach yielded an accuracy of 90.3%.

Assia Soumeur et al. [13] and [19] focused on opinions, sentiments and emotions based on various

Facebook pages' posts written in Algerian dialect. The authors applied two types of neural network models: MLP and CNN, in addition to Naïve Bayes to classify comments as negative, positive or neutral. After considering the pre-processing steps, both models achieved good accuracy results with a slightly better accuracy using the CNN model. This indicates obtaining higher accuracies using deep learning models in general.

Jalal Omer Atoum and Mais Nouman [14] focused on SA of social media users' tweets written in

Jordanian dialect. After a sequence of pre-processing steps, the dataset was labeled with positive, negative or neutral. The study applied two supervised classifiers, Naïve Bayes and SVM, on the tweets. The conducted experiments involved experimenting with different factors. The results show higher accuracy values using the SVM classifier. Results also show that using stems and root trigrams on balance data enhances the accuracy. In summary, Table 1 provides a comprehensive and comparative overview of the studied literature for the research from [5]-[14].

Table 1. A comprehensive and comparative overview of the studied literature.

Paper	Dataset	Approach	Features	Classifiers	Results (Accuracy)
[5]	Tweets Facebook comments	Supervised machine learning Lexicon-based	N-grams in words	SVM / NB K-NN	Best precision with SVM = 75.25% Best recall with K-NN = 69.04%
[6]	Tweets	Hybrid (supervised and unsupervised)	Words	SVM Naïve Bayes	F1-score of SVM = 87% F1-score of NB = 88%
[7]	LDC - ATB	Hybrid (supervised and unsupervised)	ArSenL-sentence ArSenL-Lemma Raw words	DNN DBN DAE RAE	SVM = 45.2% DNN = 39.5% DBN = 41.3 % Deep auto DBN = 43.5% RAE = 74.3%
[8]	Tweets	Hybrid (orientation and machine learning techniques)	Unigrams Bigrams Trigrams	SVM Lexical-based	F1-score = 84% Accuracy = 84.01%
[9]	Tweets	Supervised learning Lexicon-based	Higher-order N-grams (compared to unigrams up to trigrams)	Naïve Bayes SVM (LIBSVM) Baseline	F1-score of SVM = 0.384 F1-score of NB = 0.284 F1-score of baselines = 0.249 Avg. Accuracy = 0.454
[10]	Tweets	Supervised learning	TF-IDF (term frequency – inverse document frequency) Term frequency Term presence	SVM (linear kernel) Two-way, three-way, four-way classifications	Two-way: (Term presence) = 62.27% Three-way: (Term presence) =58.17% Four-way: (Term presence) = 54.69%
[11]	Facebook posts	Supervised learning	Words	Naïve Bayes SVM / DT K-NN	Best accuracy with SVM without sampling: (Avg.) 93.7% Best accuracy with SVM with sampling: (Avg.) 73.54%
[12]	Arabic online commentary (AOC)	Unsupervised learning	Lexical-based words	LSTM CNN BLSTM CLSTM	Best Accuracy = 90.3%
[13]	Facebook comments	Hybrid (supervised and unsupervised)	Word (text from comments)	MLP CNN Naïve Bayes	MLP = 81.6% CNN = 89.5% NB: Before pre-processing = 60.11% After pre-processing = 71.73%
[14]	Tweets	Supervised Semantic orientation	N-grams in word (unigrams up to N-gram)	SVM Naïve Bayes	Accuracy of NB with trigram = 55% SVM with trigram = 76%

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

In addition to previously summarized literature in Table 1, Saif M. Mohammad et al. [16] applied two different approaches to automatically generate several large sentiment lexicons. The first generating method was using distant supervision techniques on Arabic tweets and the second method was translating English sentiment lexicons into Arabic using a freely available statistical machine translation system. The authors provide a comparative analysis of the new and old sentiment lexicons in the downstream application of sentence-level SA.

Rehab Duwairi and Mahmoud El-Orfali [18] approached SA in Arabic text using three perspectives. First, investigating several alternatives for text representation; in particular, the effects of stemming feature correlation and n-gram models for Arabic text on SA. Second, investigating the behaviour of three classifiers; namely, SVM, Naïve Bayes and K-nearest neighbor, with SA. Third, analyzing the effects of the characteristics of the dataset on SA.

3. METHODOLOGY

Recently, many researchers have devoted efforts to studying the platforms of social media [30]-[31]. The interest in studying social media is due to the rapid growth of its contents as well as its impact on people's behaviour [15]. A major part of their studies focused on SA and opinion mining.

3.1 Collecting Sentiment Lexicons

In the lexicon-based approach, big efforts focused on the English sentiment lexicons [16] while little focus was placed on Arabic sentiment lexicons. On the other hand, most of these efforts focused on solving special problem statements. Arabic language sentence flow is a challenging issue due to many reasons; for example, Arabic sentences are full of using negations, modals, intensifiers and diminishers. Moreover, the Arabic language is very rich in prepositions, conjunctions, connected pronouns, object pronouns, demonstrative pronouns, relative pronouns, pronouns, paronomasia, insinuation and many other issues that require a lot of work when computerizing the Arabic language understanding. Table 2 shows Arabic language controls.

Negators change lexicons from negative to positive and the other way around. For example, the word (سعيد, happy) is a 100% positive sentiment word, but if it is preceded with any suitable negation as (غيرسعيد، ليس سعيداً، لا يعتبر سعيداً), then it is negative (not necessarily, maybe neutral).

The same is valid for the negative sentiment words if preceded with any suitable negation; for example, (ضارّ) is a 100% negative sentiment word, but when negated like (غير ضارّ، ليس ضارًّ، لا يعتبر ضارّاً), then it is positive (not necessarily, maybe neutral). Arabic negations applied are such as (لا، لم، ما، ليس، غير، لن، دون، مش مو). In diminishers, the word might be negative by itself, but in the sentence, it gives a positive connotation (كريم في الخيرات، بخيل في الإساءة الى الناس). Other examples are (حلو مُرّ لين شرس عصي طيّع). The Arabic language is rich in paronomasia words and sentences that give many meanings, like (رقيق) which has the meaning of (slave), (thin) or (gentle). The insinuations are close to paronomasias, where the speaker may say positive words, but he/she means negative ones, like (اتقوا الله).

Table 2. Arabic language controls.

Controls	Meaning in Arabic	Example in Arabic	Action
Negators	النافيات	لا، لم، ما	Manipulate
Modals	الأفعال الناقصة	أصبح، أمسى	Remove
Intensifiers	التوكيد	تكرار الكلمة	Remove
Diminishers	المتناقضات	محب مبغض	Phrases
Prepositions	حروف الجر	من، الى	Remove
Conjunctions	حروف العطف	ثم، أو	Remove
Connected Pronouns	ضمائر متصلة	الهاء في نفسه	Remove

Demonstrative Pronouns	ضمائر الإشارة	هذا، هذه	Remove
Relative Pronouns	الأسماءُ الموصُولة	الذي، التي	Remove
Pronouns	الضمائر	هو، هي	Remove
Paronomasia	التورية	أعينيَّ جودا	Phrases
Insinuation	التلميح	لا والله كريم!	Phrases

The approaches used to create Arabic sentiment lexicons can be broadly divided into three categories [32]. The first and most used approach is strongly based on the automatic translation of English sentiment lexicons and resources, either for all the entries in the lexicon or only for some parts. The second approach relies on choosing seed sentiment words and then identifying the words that occur alongside the seed words, using either statistical measures or simply using conjugates. The third approach involves human effort in manually extracting sentiment words, either from Arabic dictionaries or from datasets collected for Arabic SA and subsequently labeling these words with their polarities (positive, negative, neutral) [17].

In this work, we created Arabic sentiment lexicons through the automatic translation of English sentiment lexicons and the manual extraction of sentiment words. Next, we describe each of these resources.

3.1.1 Manually Prepared Lexicon

Our work concentrates on the comments of Facebook users related to social, public issues. The researchers had to add more words and phrases to the available lexicons because of the dialect phrases and words used by commenters as well as using negators, intensifiers, paronomasias and insinuations.

The Arabic Sentiment Lexicon comprises 333 negative phrases, 369 positive phrases, 4956 negative words and 2145 positive words, in addition to a largely manipulated negation applied on negative and positive words (39648 negative negations and 17160 positive negations). These negations were made through the concatenation of applicable negations with the sentiment's words and phrases. Figure 2 shows samples from the generated lexicons.

3.1.2 English Translated Sentiment and Emotion Lexicon

There are many English lexicons translated into Arabic, but they are hardly free of mistranslation or have different synonyms [33]-[34]. In this work, we used some lexicons from Saif M. Mohammad's collection [16]. In this collection, the author used automatic translations of English sentiment lexicons into Arabic. The study in [16] reveals that about 88% of the automatically translated entries are valid for Arabic and around 10% invalid entries were the result of gross mistranslation. 40% of the invalid entries occur due to translation into a related word and about 50% occur due to differences in how the word is used in Arabic. The translations were often the word representing the predominant sense of the word in the English source [16].

All those lexicons are very powerful in terms of Arabic sentiments words and would be helpful if researchers were to mainly analyze texts written correctly according to the Arabic linguistic structure. Unfortunately, most comments were written quickly, without correct wording (misspelling) or informed prior thinking. The Facebook post was to get the opinions of users regarding services provided by Jordanian Telecommunication Companies (Zain, Orange and Umniah) and subjects (persons) of this study who put their comments on the post seemed to be in a hurry and with no concentration when writing their comments. Most of those subjects used their own words to describe the service or to talk about their own experience with the company using sentences not always free of improvised created words, slang vocabularies and inclusion. Saif M. Mohammad and his team [16] provided huge size files that need a lot of work and approval before applying them. However, we used what seems to fit and applies to the conducted experiments.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

	POS
313	نحبك
314	نجوم
315	نجمات
316	نجم
317	نجح
318	نجتهد
319	نجاحك
320	نجاحات
321	نجاح
322	نجاه
323	نثق
324	نتفق
325	نبه
326	نبل
327	نبسط
328	نبرع
329	نبتهج
330	نبارك
331	نامى
332	ناغم
333	ناعما
334	ناعم

	POS phrases
368	يمكن الوصول
369	يمكن كسبها
370	يوجد على نحو قانوني
371	الحمد لله
372	الف مبروك
373	ارفع راسك
374	مبروك الف
375	الله الله
376	الرجل المناسب
377	ابشروا بالخير
378	اروح فدوة
379	اروح فدواك
380	نحن معكم
381	حنا معكم
382	ألف شكر
383	الام الحنون
384	الله يقويكم
385	تستحق الإشادة
386	الله يوفقكم
387	فال خير
388	فالك خير
389	اعمالهم جليلة

	NEG
1533	بطيء
1534	دنت
1535	جنائي
1536	تنقضى
1537	محتال
1538	متفجر
1539	تقشر
1540	تداعى
1541	تحطمها
1542	تألم
1543	بالذعر
1544	الهمجي
1545	المضللة
1546	كسل
1547	قصى
1548	عاقب
1549	ساذج
1550	خردة
1551	جشع
1552	جبان
1553	فوضوي
1554	عفن

	NEG phrases
4	شو مالكم
5	ضيعت فيه وقت
6	عبره لغيرك
7	عكس توقعاتي
8	اتق الله
9	اتقو الله
10	اتقى الله
11	الطين بله
12	انا صدقت بس ما
13	اهدار كرامه
14	ايام سوده
15	اكلتونا هوا
16	اكلنا هوا
17	بتعل القلب
18	بتوع مصالح
19	بلا رحمه
20	بلا معنى
21	بلا هدف
22	تضحك علينا
23	تضيع وقتك
24	تروح الاشاره

Figure 2. Sample from the lexicons.

3.2 Formulating Labeled Dataset

Related Facebook comments were collected from the Facebook[1,2,3] pages of the Jordan telecom companies with some reviews from different related comments. Figure 3 shows some sample comments from Facebook regarding the services provided by the major telecommunication companies in Jordan.

	A
1	امنية كويسة
2	امنيه ماشي حالها
3	اورانج كويسه بشكل عام
4	زين خط تلفوني كويسه بس مش ممتازه و اورنج عنا خط النت صاير سئ جدا وبطئ وعم افكر اغير بس اشوف بديل مناسب
5	خطي معمول عليه حظر بث ومشاركة ما بسمح بالبث والمشكلة فيه 10 جيجا والله احيانا بروحو باخر الشهر وعالفاضي بسبب انه الشبكة بطيئة
6	خطوط الاردن كلها تكلفتها رخيصة مقارنة بالبلدان المجاورة هاك الامارات بتكلف كثير
7	اتصالات زين على البلدان المجاورة مثل السعودية تكلفتها منطقية ومناسبة
8	اسخف عروض واسخف خدمات شفتها بحياتي
9	الحقيقة انه احسن من هيك شركة ما في
10	الخدمة مريحة والتكلفة مناسبة سرعة الانترنت مقبولة
11	الافضل والاحسن والاقوى شركة اورانج ولا منازع
12	هيك الشركات ولا بلاش انا جربت كل الشركات والله تعامل ما شاء الله قمة الاحترام
13	زين بحسه أفضل شي بعد عدة تجارب
14	زين والسبب تغطية اوسع على مستوى المملكة
15	على الرغم من انه عروض الشبكات الاخرى افضل ولكن شو الفايدة من انه يكون معك بيانات اكثر ونسبة التحميل ضعيفة
16	ملاحظة احيانا تجد في بعض المناطق اورانج او امنية اقوى ولكنها مناطق محدودة
17	زين مكالمات ونت ممتاز بس بسحبو نت كتير
18	منيه بس طبعا حسب المنطقة قوة او ضعف النت وكل الشبكات نفس المشكلة
19	اورانج الشبكة ممتازة و الزين ممتازة كمان
20	امنية وكثير بيجنن والحزم بقعدو معي شهر كامل والشبككه ممتازه

Figure 3. Sample from the collected comments.

[1] https://m.facebook.com/OrangeJordan
[2] https://m.facebook.com/zainjordan
[3] https://m.facebook.com/Umniah

After loading and labelling the collected comments, the researchers applied some text pre-processing on them. Pre-processing is a vital step for getting sentimental text. The main tools that were used in this step are Microsoft Excel and KNIME. Microsoft Excel was used in preparing and proofing the lexicon words, concatenation of negations with lexicon words, removing modals, prepositions, pronouns and intensifiers, as well as other text operations, like refinement and trimming of demonstrative pronouns.

Meanwhile, KNIME was used in the analysis and labeling. The main pre-processing steps that were performed are the following steps:

- Removing punctuations including apostrophe, colon, comma, dash, ellipsis, exclamation point, hyphen, parentheses, period, question mark, quotation mark and semicolon.
- Replacing some letters and words with other letters or words that are known to the system. Here are some examples: (أ/آ/إ/اًby ا), (ئ/ىby ي), (ليست/ليستا/لستما/ليسا/لسنا//لستم لستن /ليسوا/لست/لسن by ليس), (ةby ه), (ؤby و), removing (الـ) by replacing it with space, replacing (لا يعتبر/لا يعد by ليس).
- Removing of propositions, such as: (من، الى، عن، على، في).
- Tagging positive phrases, like (الف مبروك، والله نشامى) and negative words, such as: (اتقوا الله، حسبي الله فيكم، عديم الفائدة).
- Tagging of negations.
- The priority of tagging was as follows: negative phrases, positive phrases, negated negative, negated positive, negative and positive. Note that after applying Dictionary Tagger in KNIME on a phrase or word, it will not be changed. Mostly, the researchers focused on negative phrases and words, since this solution follows the lexicon-based approach to perform sentiment analysis. A recent and important work can be referred to in [2].
- Filtering the tagged words, so that only sentiment words are included in the counting of sentiments.
- Creating a bag of words that separate each group of sentiment words individually.
- Counting the frequency of each group.
- Calculating the result using Equation 1 and if the value gained from Equation 1 is greater than or equal the mean (result), then the comment is POSITIVE; otherwise, the comment is NEGATIVE.

$$Results = \frac{POS_{words} - NEG_{word}}{TOTAL_{words}} \tag{1}$$

Figure 4 represents the previously mentioned steps for pre-processing, analyzing and labelling of collected comments, then formulating a labeled dataset.

In this research, we have applied the KNIME Analytics Platform for analysis and labeling based on the dictionary tagger. KNIME is a software built for fast, easy and intuitive access to advanced data science, helping organizations drive innovation. KNIME[4] Analytics Platform became one of the leading open solutions for data-driven innovation, designed for discovering the potential hidden in data, mining for fresh insights or predicting new features. Figure 5 shows the KNIME analyzing and labeling model.

From the model in Figure 5, KNIME parts, nodes, extensions and components of the model along with its purpose are as follows:

- "Excel Reader (XLS)" node is designed for reading Text Data from Excel files. Here, we manually collected the dataset from three Facebook pages. Then, we inserted it into the Excel sheet in addition to other lexicon files.
- "Strings to Document" node converts the specified strings into documents. The input is a data table containing string cells and the output is a table containing the strings of the data of the input table as

[4] Knime official website: https://www.knime.com/about

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

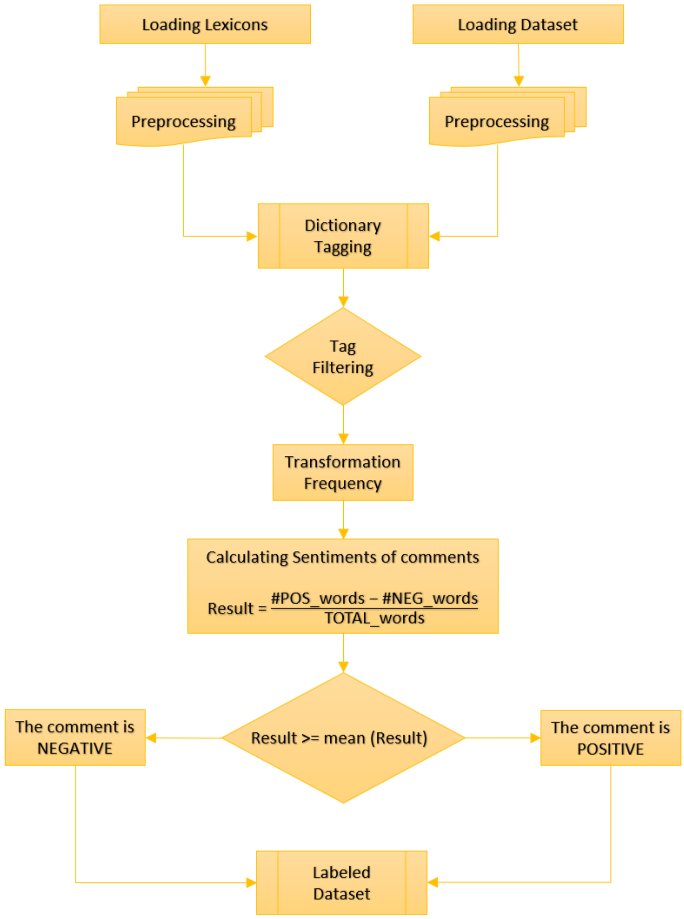


Figure 4. Labeling dataset and pre-processing flow.

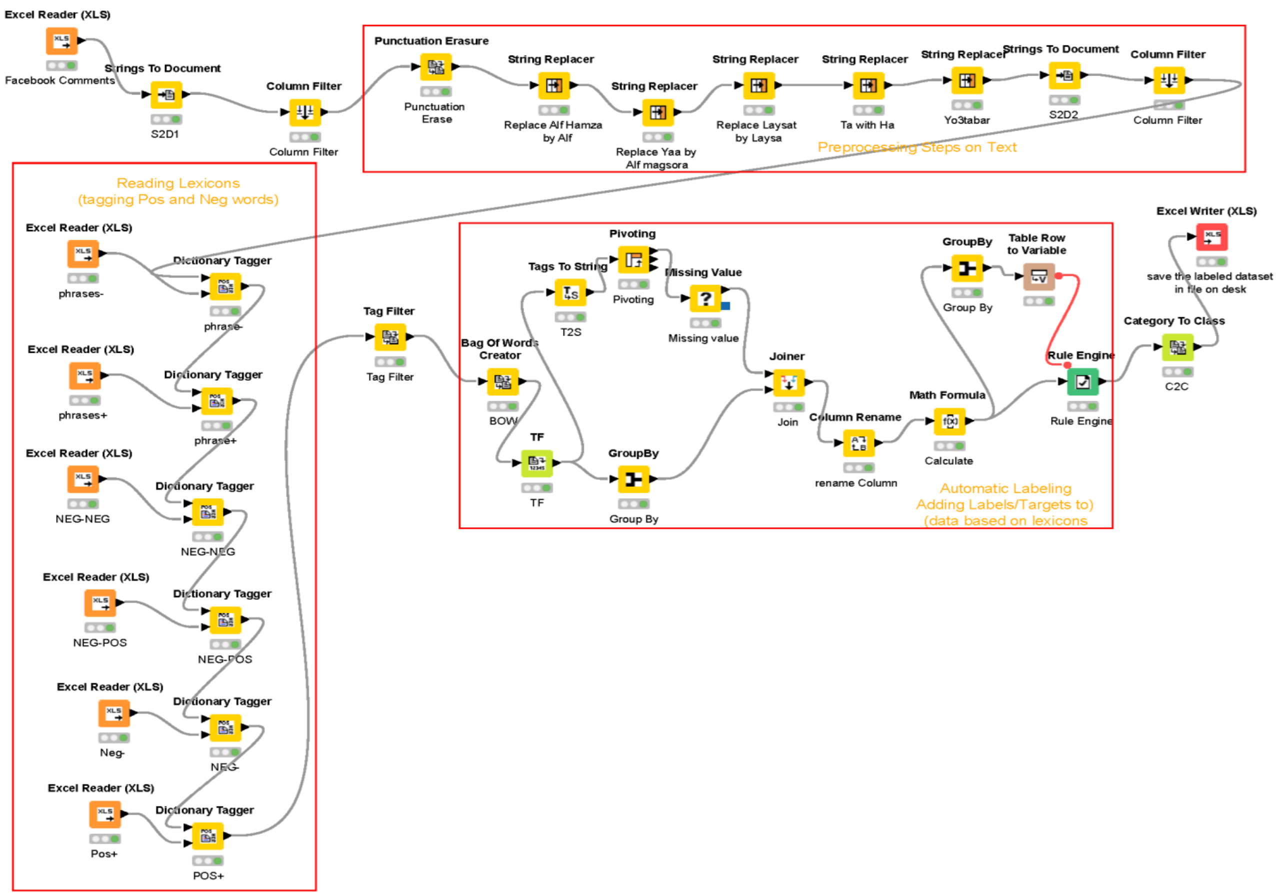


Figure 5. Analyzing and Labeling in KNIME.

well as the created documents in an additional column.

- "Column Filter" node filters certain columns from the input table and allows the remaining columns to pass through the output table.
- "Punctuation Erasure" node removes all punctuation characters of terms contained in the input documents. Input is the table that contains the documents before pre-processing and output is the table that contains the documents after pre-processing.
- "String Replacer" node is for replacing and removing pre-positions and other punctuations previously listed. It replaces values in string cells if they match a certain wildcard pattern. Input is arbitrary data and the output column contains the SA (NEG, POS) of each data column (comment). The SA results are calculated using Equation 1.

The labeling process in this manner was unsupervised. It was depending on the collected data, either from Arabic lexicons or English translated lexicons after refinement and filtering.

4. Supervised Machine Learning

The previously described workflow in section 3 resulted in labeled data (unsupervised). To test the accuracy of the proposed approach, we used the labeled dataset to formulate a Machine Learning model using ORANGE software to implement the classifier. ORANGE is an open-source data visualization, machine learning and data mining toolkit. It features a visual programming front-end for explorative data analysis and interactive data visualization. ORANGE is a component-based visual programming software package for data analysis. ORANGE's components are called widgets and they range from simple data visualization, subset selection and pre-processing, to empirical evaluation of learning algorithms and predictive modeling. In ORANGE, visual programming is implemented through an interface in which workflows are created by linking predefined or user-designed widgets, while advanced users can use ORANGE[5] as a Python library for data manipulation and widget alteration.

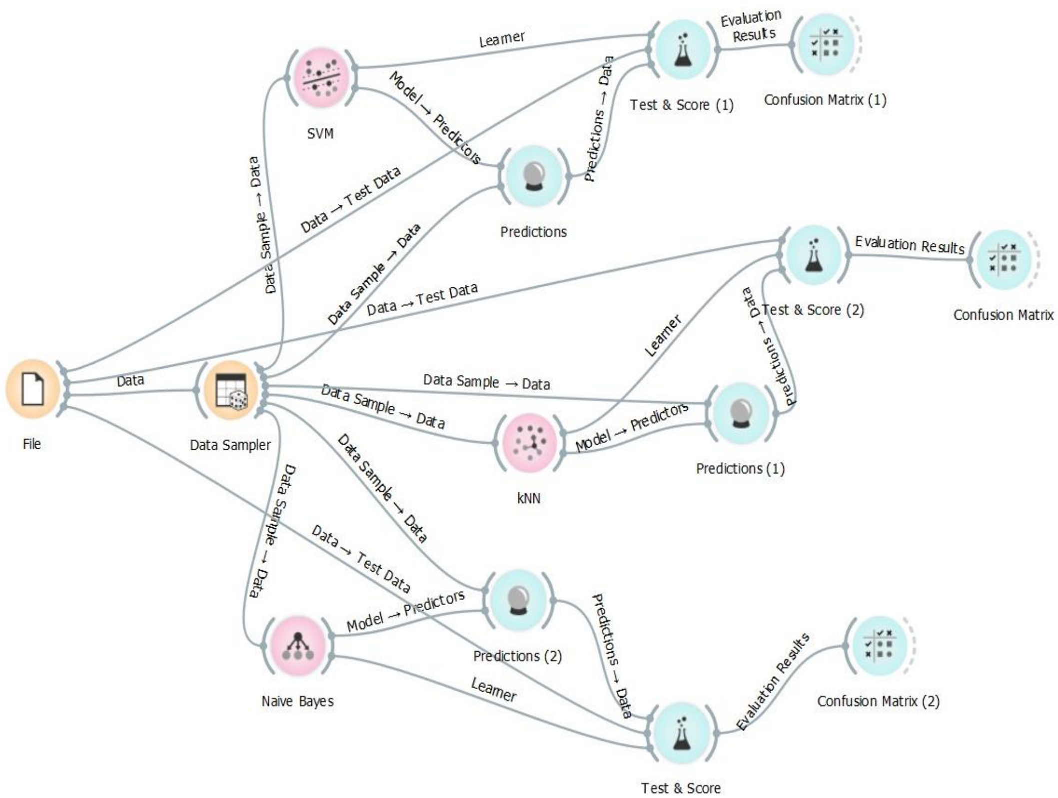


Figure 6. ORANGE-workflow (ML model of SVM, NB and K-NN).

[5] ORANGE official website: https://orange.biolab.si/workflows/

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

Experiments were conducted on the 1300 comments to produce a Predictive Machine Learning Model (PMML) (supervised) using the PMML predictor. A text pre-processing step is necessary to remove all unnecessary and misleading words. Then, we experiment with the pre-processed text on Machine Learning. Figure 6 shows the workflow of the ML model implemented using ORANGE software. After that, we tested the collected dataset using SVM, NB and K-NN algorithms, since they are the most used algorithms in this context.

In summary, Figure 7 illustrates our proposed work from unsupervised labeling using a lexicon-based approach till supervised learning verification of the labeled comments using the ML model.

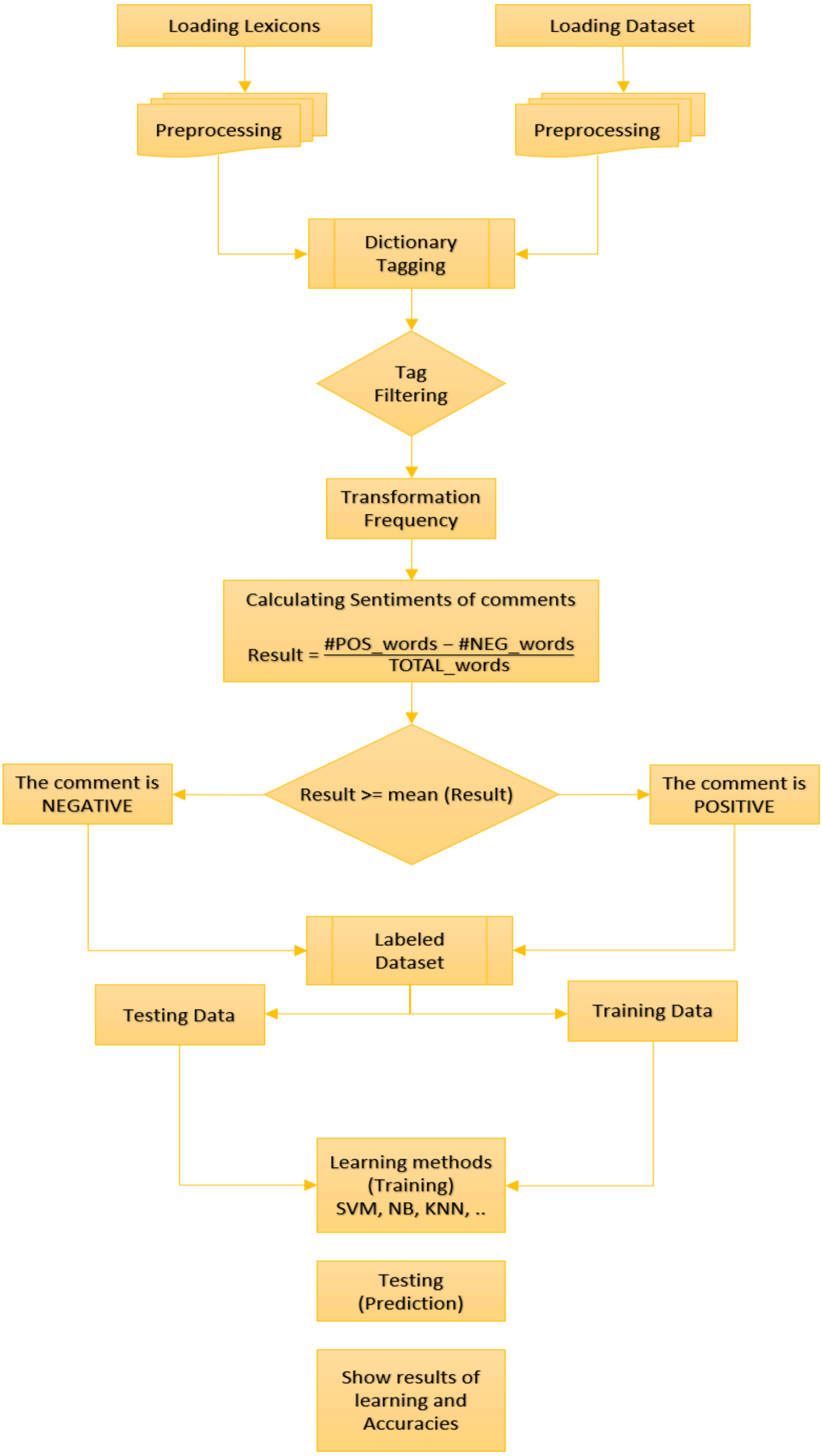


Figure 7. Methodology flow diagram.

5. RESULTS

This paper has addressed SA in Arabic comments. First, we implemented the lexicon-based approach for identifying the polarity of the provided text. Lexicons were two types of pure Arabic lexicons and refined and filtered English translated lexicons. The data samples are from local Jordanian people commenting on a public issue related to the services provided by the main telecommunication companies in Jordan. Second, we used the resulting labeled dataset frequently used ML algorithms for classification of comments in the absence of lexicons. The workflow in Figure 7 shows the whole process starting with importing the data through labelling it and ending with classification and results in SA. The procedure involves applying a user-defined lexicon based on the common Facebook posts and comments used by Jordanians, which resulted in a (60%) positive comments and (40%) negative ones. The total accuracy of lexicon-based labeling was calculated through a comparison between the achieved results and the ones achieved through manually labeled comments by experts. The general accuracy of lexicon-based labeling was (98%). Higher accuracy values can be accomplished by adding more words and phrases to our lexicons.

Figure 8 shows a sample of the results. The SA column (F) is automatically produced using the KNIME Software nodes (Tag Filter node in Figure 5). Researchers may see the partial results comment by comment with their sentiments using (Document view node).

	A	B	C	D	E	F
1	comment	pos words	neg words	total words	result	sentiment
2	""امنيه وخط متميز نت واتصالات عل كل الشبكات مرتاح كثير""	4	0	4	1	POS
3	""امنيتى ان يحيا اطفال العالم والمسلمين وخاصه ابناء غزه بسلام""	4	0	4	1	POS
4	""امنيه - جيده""	2	0	2	1	POS
5	""امنيه اتوقع اسوا خط بالعالم""	0	2	2	-1	NEG
6	""امنيه احسن عروض""	2	0	2	1	POS
7	""امنيه النت ضعيف كثير""	0	2	2	-1	NEG
8	""امنيه النت فيه بمنطقتى شبه عدم مابستخدمه نهائى بستخدم النت الارضي""	0	4	4	-1	NEG
9	""امنيه النت كتير سيئ بس يخلص الاشتراك ما برجعلهم""	0	4	4	-1	NEG
10	""انه لشيء جميل الاشتراك بخدمات زين المميزه""	2	0	2	1	POS
11	""اورانج ،،، اسوا نت فايبر""	0	2	2	-1	NEG
12	""اورانج الخدمه سيئه جدا وكتير وقدمنا شكوى اكتر من مره وما تغير عليها شي""	4	4	8	0	NEG
13	""اورانج الشبكه ممتازه و الزين ممتازه كمان""	4	0	4	1	POS
14	""اورانج النت بطيء""	0	2	2	-1	NEG
15	""اورانج بس النت قليل""	0	2	2	-1	NEG
16	""اورانج بس سيئه جدا من ناحيه حزم البيانات سرقه عينك عينك مشكوووك فيها هالشركه""	2	4	6	-0.33	NEG
17	""اورانج كتير منيح خدمات وعروض واشتراكات فواتير شهريه ونطاق النت سريـع""	4	0	4	1	POS
18	""اورانج كثير كويس""	2	0	2	1	POS
19	""اورانج كويسه بشكل عام""	4	0	4	1	POS
20	""بس النت ضعيف جدا جدا جدا""	0	2	2	-1	NEG
21	""بس لو النت ما يخلص بسرعه""	2	2	4	0	NEG
22	""زين انت مثل الكحل للعين""	2	0	2	1	POS
23	""زين بعض الاحيان تكون ضعيفه وخاصه شبكه النت""	0	2	2	-1	NEG
24	""زين خدماتهم حلوه بس بالنت الويرلس فاشل""	2	2	4	0	NEG
25	""زين شايفته كويس من ناحيه التغطيه وسرعه النت لكن بحسهم نصابين بحزم البيا	4	0	4	1	POS

Figure 8. Sample results of labeling.

1	Comments	Pos-Words	Neg-Words	Total Words	Results	Sentiment
2	اقسم بالله لولا اشتراكي السنوي ولا الان بفصل اشتراكي بس المشكلة اني دفعت لسنة كاملة اقرف من هيك خدمة ما في والنت بطيء وسيء والحقيقة اني خلص اكتفيت منهم	2	10	12	-0.66666667	?
3	الخدمات ممتازة عروض حلوة ومفيدة تكلفة ما شاء الله مريحة مع اورانج	10	0	10	1	?
4	النت قوي وسريـع لكن بساعات المساء تضعف قليلا	8	2	10	0.6	?
5	انا بشكر شركة زين لانها تحفظ حق المستخدم فعلا	6	0	6	1	?
6	افضل شركة هي امنية	2	0	2	1	?
7	امنية والله العكس تماما بطيء جدا تغطية معدومة	0	8	8	-1	?
8	امنية عديمة الجدوى على الفاضي بس شغل سحب مصاري الله لا يربحهم بشيء	6	2	8	0.5	?
9	8 اشهر وانا انتظر عرض ينفع والان لقيت العرض 1000 جيجا ب12 دينار والله مفخرة	4	0	4	1	?
10	كويس جدا خط اورانج يعطي نت 5 جيجا استخدام شخصي مريـح جدا والسرعة ممتازة	6	0	6	1	?
11	اتمنى انكم تعملو تحديث للعروض يعني هيك اشي ب3 دنانير يكون 15 جيجا مثلا ويكون يعمل مشاركة اهم اشي واشكركم مقدما اقتراح بس	4	0	4	1	?

Figure 9. Sample of unlabeled comments.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

In Figure 8 and by referring to Figure 5, column A is the comment, columns B and C are sentiments that have resulted from the (TF node). This data is grouped (Group By) and columns are pivoted (Pivoting) to produce column D (total sentiments). The (Math formula node) is calculating the result (E) and finally with (Rule Engine node) a new column is added (F), which is our targeted LABELING (Sentiment of the comment: POS or NEG). Figure 9 shows a sample of unlabeled comments provided to the lexicon-based model, whereas the labeling results (SA) are shown in Figure 10.

1	Comments	Pos-Words	Neg-Words	Total Words	Results	Sentiment
2	اقسم بالله لولا اشتراكي السنوي ولا الان بفصل اشتراكي بس المشكلة اني دفعت لسنة كاملة اقرف من هيك خدمة ما في والنت بطيء وسيء والحقيقة اني خلص اكتفيت منهم	2	10	12	-0.66666667	NEG
3	الخدمات ممتازة عروض حلوة ومفيدة تكلفة ما شاء الله مريحة مع اورانج	10	0	10	1	POS
4	النت قوي وسريع لكن بساعات المساء تضعف قليلا	8	2	10	0.6	NEG
5	انا بشكر شركة زين لانها تحفظ حق المستخدم فعلا	6	0	6	1	POS
6	افضل شركة هي امنية	2	0	2	1	POS
7	امنية والله العكس تماما بطيء جدا تغطية معدومة	0	8	8	-1	NEG
8	امنية عديمة الجدوى على الفاضي بس شغل سحب مصاري الله لا يربحهم بشيء	6	2	8	0.5	NEG
9	8 اشهر وانا انتظر عرض ينفع والان لقيت العرض 1000 جيجا ب12 دينار والله مفخرة	4	0	4	1	POS
10	كويس جدا خط اورانج يعطي نت 5 جيجا استخدام شخصي مريح جدا والسرعة ممتازة	6	0	6	1	POS
11	اتمنى انكم تعملو تحديث للعروض يعني هيك اشي ب3 دنانير يكون 15 جيجا مثلا ويكون يعمل مشاركة اهم اشي واشكركم مقدما اقتراح بس	4	0	4	1	POS

Figure 10. Unlabeled comments after labeling.

About 1300 comments were collected from Facebook pages and provided to the lexicon model for labeling. The accuracy achieved was 98% based on some experts and by expressing some comparisons with other labeled comments. Unfortunately, this model is still restricted to the availability of the words or phrases in the lexicons. It is considered as unsupervised learning that depends on a mathematical counting formula.

On the other hand, researchers can use the resulting labeled dataset to build an ML model that will efficiently classify any newly outlet comments, which is considered a supervised learning model. To perform the classification, we applied three classifiers; namely, SVM, NB and K-NN using ORANGE software tool. The model for the three famous classifiers and its details are shown in Figure 6. The accuracy results were very promising. Table 3 shows the accuracy results of those classifiers. Table 3 makes clear that all classifiers provided good results, but superiority was for the SVM classifier since it is powerful when dealing with binary classification problems.

Table 3. Classification results.

Classifier	**Accuracy**
Support Vector Machine (SVM)	**97.9 %**
k-nearest neighbor (K-NN)	**96.8 %**
Naïve Bayes (NB)	**95.6 %**

6. Conclusions

This paper has focused on SA of Facebook Arabic comments for Jordanian telecom companies. The output of our work was an Arabic Sentiment Lexicon, which comprises 333 negative phrases and 369 positive phrases. Besides, the researchers have collected 4956 negative words and 2145 positive words in addition to a largely manipulated negation applied to negative and positive words. Most of the phrases and words came from the Jordanian dialect and MSA in addition to the applicable sentiment words from the English sentiments translated into Arabic.

The researchers implemented the lexicon-based approach for identifying the polarity of each of the provided Facebook comments. Data samples are from local Jordanian people commenting on a public issue related to the services provided by the main telecommunication companies in Jordan (Zain, Orange and Umniah). The produced results regarding the evaluation of Arabic sentiment lexicon were promising. When applying the user-defined lexicon based on the common Facebook posts and comments used by Jordanians, it scored (60%) positive and (40%) negative. The general accuracy of the lexicon was (98%). The lexicon was used to label a set of Facebook comments to formulate a big dataset of unlabeled comments. Using supervised Machine Learning (ML) algorithms that are usually used in polarity classification, the researchers introduced them to our formulated dataset. The results of the classification were 97.8, 96.8 and 95.6% for Support Vector Machine (SVM), K-Nearest Neighbour (K-NN) and Naïve Bayes (NB) classifiers, respectively. It is worthy to note that without applying Arabic language grammar rules and Arabic sentence structure, any lexicon would fail in such a task because of issues related to the Arabic language.

7. Future work

The formulated lexicons can be improved by adding new phrases and words related to sentiments that will improve the accuracy and quantity of labeling. The paper highlights the need to have a dedicated website for uploading lexicons and datasets collected by researchers in the field of NLP which may be helpful in this context. Moreover, there are other fields in NLP that rely on the lexicon approach, which makes this work exploited in other tasks. To overcome some of the challenges of Arabic sentiment analysis, we are considering the use of recourses as SenticNet [23].

References

[1] M. D. Devika, C. Sunitha and A. Ganesh, "Sentiment Analysis: A Comparative Study on Different Approaches," Procedia Comput. Sci., vol. 87, no. June, pp. 44–49, 2016.

[2] B. Liu, Sentiment Analysis (Introduction and Survey) and Opinion Mining, [Online], Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.244.9480&rep=rep1&type=pdf, 2012.

[3] R. K. Bakshi, N. Kaur, R. Kaur and G. Kaur, "Opinion Mining and Sentiment Analysis," Proc. of the IEEE 3rd Int. Conf. Comput. Sustain. Glob. Dev. (INDIACom), vol. 2, no. 1, pp. 452–455, 2016.

[4] A. Chandra Pandey, D. Singh Rajpoot and M. Saraswat, "Twitter Sentiment Analysis Using Hybrid Cuckoo Search Method," Inf. Process. Manag., vol. 53, no. 4, pp. 764–779, 2017.

[5] R. M. Duwairi and I. Qarqaz, "Arabic Sentiment Analysis Using Supervised Classification," Proc. of the 1st International Workshop on Social Networks Analysis, Management and Security (SNAMS - 2014), pp. 579–583, Barcelona, Spain, August 2014.

[6] R. M. Duwairi, "Sentiment Analysis for Dialectical Arabic," Proc. of the 6th Int. Conf. Inf. Commun. Syst. (ICICS), DOI: 10.1109/IACS.2015.7103221, pp. 166–170, Amman, Jordan, 2015.

[7] A. Al Sallab, H. Hajj, G. Badaro, R. Baly, W. El Hajj and K. Bashir Shaban, "Deep Learning Models for Sentiment Analysis in Arabic," Proceedings of the Second Workshop on Arabic Natural Language Processing, DOI: 10.18653/v1/W15-3202, pp. 9–17, Beijing, China, 2015.

[8] H. K. Aldayel and A. M. Azmi, "Arabic Tweets Sentiment Analysis - A Hybrid Scheme," J. Inf. Sci., vol. 42, no. 6, pp. 782–797, 2015.

[9] H. Mulki, H. Haddad, M. Gridach and I. Babaoğlu, "Tw-StAR at SemEval-2017 Task 4: Sentiment Classification of Arabic Tweets," Proc. of the 11th International Workshop on Semantic Evaluations (SemEval-2017), DOI: 10.18653/v1/S17-2110, pp. 664–669, Vancouver, Canada, 2018.

[10] N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman and Y. Al-Ohali, "AraSenTi-Tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets," Procedia Comput. Sci., vol. 117, pp. 63–72, 2017.

[11] H. Najadat, A. Al-Abdi and Y. Sayaheen, "Model-based Sentiment Analysis of Customer Satisfaction for the Jordanian Telecommunication Companies," Proc. of the 9th International Conference on Information and Communication Systems (ICICS 2018), DOI: 10.1109/IACS.2018.8355429, 2018.

[12] L. Lulu and A. Elnagar, "Automatic Arabic Dialect Classification Using Deep Learning Models," Procedia Comput. Sci., vol. 142, pp. 262–269, 2018.

[13] A. Soumeur, M. Mokdadi, A. Guessoum and A. Daoud, "Sentiment Analysis of Users on Social Networks: Overcoming the Challenge of the Loose Usages of the Algerian Dialect," Procedia Computer Science, vol. 142, pp. 26-37, 2018.

[14] J. O. Atoum and M. Nouman, "Sentiment Analysis of Arabic Jordanian Dialect Tweets," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 2, pp. 256–262, 2019.

[15] M. Bouaziz and T. Ohtsuki, "Multi-class Sentiment Analysis in Twitter: What If Classification Is Not the Answer," IEEE Access, vol. 6, pp. 64486-64502, 2018.

[16] S. Kiritchenko, S. M. Mohammad and M. Salameh, "SemEval-2016 Task 7: Determining Sentiment Intensity of English and Arabic Phrases," Proc. of the 10th International Workshop on Semantic Evaluation (SemEval-2016), DOI: 10.18653/v1/S16-1004, San Diego, California, 2016.

[17] A. Al-Thubaity, Q. Alqahtani and A. Aljandal, "Sentiment Lexicon for Sentiment Analysis of Saudi Dialect Tweets," Procedia Computer Science, vol. 142, pp. 301-307, 2018.

[18] R. M. Duwairi and M. El-Orfali, "A Study of the Effects of Pre-processing Strategies on Sentiment Analysis for Arabic Text," Journal of Information Science, vol. 40, no. 4, pp. 501–513, 2014.

[19] M. Mataoui, O. Zelmati and M. Boumechache, "A Proposed Lexicon-based Sentiment Analysis Approach for the Vernacular Algerian Arabic," Res. Comput. Sci., vol. 110, pp. 55-70, 2016.

[20] S. L. Lo, E. Cambria, R. Chiong et al., "Multilingual Sentiment Analysis: From Formal to Informal and Scarce Resource Languages," Artif. Intell. Rev., vol. 48, pp. 499–527, 2017.

[21] O. Oueslati, A. I. S. Khalil and H. Ounelli, "Sentiment Analysis for Helpful Reviews Prediction," International Journal of Advanced Trends in Computer Science and Engineering, vol. 7, no. 3, pp. 34-40, DOI: 10.30534/ijatcse/2018/02732018, 2018.

[22] E. Cambria, "Affective Computing and Sentiment Analysis," IEEE Intelligent Systems, vol. 31, no. 2, pp. 102-107, DOI: 10.1109/MIS.2016.31, 2016.

[23] E. Cambria, S. Poria, D. Hazarika and K. Kwok, "SenticNet 5: Discovering Conceptual Primitives for Sentiment Analysis by Means of Context Embeddings," Proc. of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), pp. 1795–1802, New Orleans, Louisiana, USA, 2018.

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

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

[26] A. Al-Shaikh, R. Al-Sayyed and A. Sleit, "A Case Study for Evaluating Facebook Pages with Respect to Arab Mainstream News Media," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 3, no. 3, December 2017.

[27] A. Bouziane, D. Bouchiha, N. Doumi and M. Malki, "Toward an Arabic Question Answering System over Linked Data," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 4, no. 2, August 2018.

[28] E. Al-Shawakfa, "A Rule-based Approach to Understand Questions in Arabic Question Answering," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 2, no. 3, Dec. 2016.

[29] Oueslati et al., "A Review of Sentiment Analysis Research in Arabic Language," Future Generation Computing Systems, vol. 112, pp. 408-430, 2020.

[30] K. Nahar, N. Alhindawi, O. Al-Hazaimeh, R. M. Al-Khatib and A. Al-Akhras, "NLP and IR-based Solution for Confirming Classification of Research Papers," Journal of Theoretical and Applied Information Technology, vol. 96, pp. 5269-5279, 2018.

[31] M. Alshanaq, K. Nahar and K. Halawani, "AQAS: Arabic Question Answering System Based on SVM, SVD and LSI," Journal of Theoretical and Applied Information Technology, vol. 97, pp. 681-691, 2019.

[32] K. Nahar, M. Alshanaq, 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, pp. 458-464, 2019.

[33] K. Nahar, A. Al Eroud, M. Barahoush and A. Al-Akhras, "SAP: Standard Arabic Profiling Toolset for Textual Analysis," International Journal of Machine Learning and Computing, vol. 9, pp. 222-229, 2019.

[34] K. Nahar, R. Al-Khatib, M. Alshanaq, M. Daradkeh and R. Malkawi, "Direct Text Classifier for Thematic Arabic Discourse Documents," International Arab Journal of Information Technology, vol. 17, (Published Online), 2020.

ملخص البحث:

تحليل الآراء إحدى التقنيات التي تستخدم لتحديد القطبية (موجب، سالب) لنصّ معين باستخدام تقنيات معالجة اللغات الطبيعية. ويُعدّ فيسبوك مثالاً على منصّات التواصل الاجتماعي الأوسع استخداماً بين المواطنين والمقيمين في الأردن للتعبير عن آرائهم بخصوص قضية عامة محوريّة.

في هذه الورقة، تم استخدام منهجية قائمة على المعاجم من أجل تحديد قطبية عدد من التعليقات المنشورة على فيسبوك. البيانات الخاضعة للتحليل هي لأشخاص أردنيين يعلّقون على قضية عامّة تتعلق بالخدمات التي تُقدمها شركات الاتصالات الرئيسية العاملة في الأردن (زين، وأورانج، وأمنية). وكانت النتائج المتعلقة بتقييم معجم الآراء باللغة العربية واعدة. وبتطبيق المعجم القائم على البوستات والتعليقات المنشورة على فيسبوك التي يعبر بها المستخدمون الأردنيون عن آرائهم، مثلث التعليقات الإيجابية ما نسبته 60% والتعليقات السلبية ما نسبته 40%، وكانت النسبة العامة لدقة المعجم 98%.

وقد استخدم المعجم لوسْم مجموعة من التعليقات غير الموسْومة على فيسبوك لتشكيل مجموعة بيانات كبيرة. وباستخدام عدد من خوارزميات تعلُّم الآلة التي يشيع استخدامها لتحديد القطبية، تم تطبيق هذه الخوارزميات على مجموعة البيانات التي جرى تشكيلها في هذه الدراسة. وكانت نتائج التصنيف 97.8%، و96.8%، و95.6% لمصنِّفات (SVM، وK-NN، وNB) على الترتيب.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

A SCALABLE SHALLOW LEARNING APPROACH FOR TAGGING ARABIC NEWS ARTICLES[1]

Leen Al Qadi, Hozayfa El Rifai, Safa Obaid and Ashraf Elnagar

(Received: 28-Mar.-2020, Revised: 23-Jun.-2020 and 15-Jul.-2020, Accepted: 16-Jul.-2020)

ABSTRACT

Text classification is the process of automatically tagging a textual document with the most relevant set of labels. The aim of this work is to automatically tag an input document based on its vocabulary features. To achieve this goal, two large datasets have been constructed from various Arabic news portals. The first dataset consists of 90k single-labeled articles from 4 domains (Business, Middle East, Technology and Sports). The second dataset has over 290k multi-tagged articles. The datasets shall be made freely available to the research community on Arabic computational linguistics. To examine the usefulness of both datasets, we implemented an array of ten shallow learning classifiers. In addition, we implemented an ensemble model to combine best classifiers together in a majority-voting classifier. The performance of the classifiers on the first dataset ranged between 87.7% (Ada-Boost) and 97.9% (SVM). Analyzing some of the misclassified articles confirmed the need for a multi-label opposed to single-label categorization for better classification results. We used classifiers that were compatible with multi-labeling tasks, such as Logistic Regression and XGBoost. We tested the multi-label classifiers on the second larger dataset. A custom accuracy metric, designed for the multi-labeling task, has been developed for performance evaluation along with hamming loss metric. XGBoost proved to be the best multi-labeling classifier, scoring an accuracy of 91.3%, higher than the Logistic Regression score of 87.6%.

KEYWORDS

Arabic text classification, Single-label classification, Multi-label classification, Arabic datasets, Shallow learning classifiers.

1. INTRODUCTION

Large numbers of online repositories have been created continuously in the recent decades, due to the non-stop flow of information, as well as the heavy usage of the internet and Web 2.0. This increase of online documents was followed by a growing demand for automatic categorization algorithms. 80% of this information is in an unstructured form and the most common type is the “textual” data. Although it is considered to be an extremely rich source of information, it becomes harder to extract insights from or deduce trends when it’s presented in enormous amounts. Machine learning techniques are often used to organize massive chunks of data and perform a number of automated tasks.

Natural Language Processing (NLP) [2], is a field of study concerned with analyzing and processing natural language data in large amounts. Machine learning algorithms, in addition to deep learning methods, are used in NLP to fulfil several tasks like the text classification task. It is the task of classifying text and assigning it appropriate tags, based on its content. The act of classification will standardize the platform, make the process of searching information easier and more feasible and simplify the overall experience of automated navigation.

Structuring data is also useful in the world of business and organizations. It will enhance the decision-making, identify current trends and predict new ones. In addition to automating regular processes, marketers can research, collect and analyze keywords by competitors.

Manual classification performed by experts is not always fruitful or efficient, due to human errors and to the long time needed to do it. Using machine learning as an alternative is proving to be more effective and, in some cases, more precise. Applications of text classification have been explored, such

---

[1] This paper is an extended version of a short paper [44] that was presented at the 2nd International Conference "New Trends in Information Technology (ICTCS)", 9-11 October 2019, Amman, Jordan.

[2] Natural Language Processing https://monkeylearn.com/natural-language-processing/

L. Al Qadi, H. El Rifai, S. Obaid and A. Elnagar are with Computer Science Department, University of Sharjah, UAE. Emails: u16103630@sharjah.ac.ae, u16103377@sharjah.ac.ae, u16103014@sharjah.ac.ae and ashraf@sharjah.ac.ae

as sentiment analysis [1]-[6], spam filtering [7]-[8], language identification [9], dialect identification [10] and many more.

Some languages present big challenges to many NLP applications [11] and the Arabic language is one of them. It is the mother tongue of over 300 million people and it is considered to be a significantly inflected and derived language. Compared to the English language, the scale of computational linguistic research on the Arabic language is relatively small but is now much bigger than what was available about a decade ago.

The Internet World stats reports that the Arabic language is the 4th most popular language among online users, with an estimate of 226,595,470 Arabic users. As of April 2019, that number represents 5.2% of the world's internet users. Additionally, it shows that 51.0% of all Arabic speaking people (in 2019) use the internet and that the language has the highest growth rate, of online users, among all the languages in the last 19 years, scoring a percentage of 8,917.3%.

In this context, we present a newly constructed Arabic news articles dataset, collected by scraping a number of websites for the purpose of our research. We implement 10 classical classifiers to predict the most suited category for a certain news article. Moreover, we design a voting classifier that classifies an article with respect to the output of selected models.

A developed system for Arabic news article single-labeling will extract text features from the text using the Tf-IDF technique. In the training phase, all the articles are turned to feature vectors, which will identify the common features that define each class separately. After the model is trained on the features, it will easily predict the class of an article after being vectorized.

To label news articles under one of 4 classes, we propose a single-class classifier, using a supervised machine learning approach. Two different vectorization methods were tested and compared to observe the effects of using each on the accuracy of the models. Lastly, the effects of using a custom-made stop words list in place of the built-in list provided by the NLTK library were also explored.

A decision to build a new multi-labeled Arabic dataset is made after analyzing the misclassified articles, classified by the single-class classifier. The need to assign multiple tags to an article instead of a single one was apparent and 2 classical classifiers were implemented for the task. The Tf-IDF technique was also used to extract the linguistic features. To decompose the multi-labeling problem into independent binary classification problems, we wrapped each of the classifiers in a OneVsRest classifier.

With an objective to assign articles to multiple labels (out of 21 labels), we propose a multi-label text classifier. We test the classifiers and evaluate them using a custom accuracy metric, along with comparing the hamming-loss scores.

The remaining of the paper is organized as follows: literature review is presented in Section 2. Section 3 demonstrates the datasets. Section 4 describes the proposed classification systems. Section 5 presents the experimental results. Finally, we conclude the work in Section 6.

2. LITERATURE REVIEW

Several papers review the various English text classification approaches and existing literature in addition to the many surveys covering the subject [12]-[14]. Some surveys that cover Arabic text categorization are also available [15]-[16].

Light has been shed on the research papers that focused on using the classical supervised machine learning classifiers, such as Decision Tree [17]-[19], NB [20]-[23], SVM [19], [22], [26]-[27] and KNN [23], [26], while other authors preferred to explore text classification using deep learning and neural networks [25] and some also witnessed an overall better performance [27]-[35].

Recently, more research works are focusing on Arabic text classification and on enriching the Arabic corpus. In [36], the authors have compared the results of using six main classifiers, using the same datasets and under the same environmental settings. The datasets were mainly collected from (www.aljazeera.net) and it was found that Naive Bayes gave the best results, with or without using feature selection methods.

Other papers focus on the feature selection method, like in [37]. Implementing the KNN classifier, the

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

authors studied the effect of using unigrams and bigrams as representation of the documents, instead of the traditional single-term indexing (bag of words) method. Moreover, on feature selection, in [38], the authors investigated the performance of four classifiers using 2 different feature-selection methods which are Information Gain (IG) and the ($X2$) statistics (CHI squared) on a BBC Arabic dataset. The use of SVM classifier (with Chi squared feature selection) for Arabic text classification, in [39], gives the best results. In [40], a new feature selection method is presented, where it outperformed five other approaches using the SVM classifier.

Regarding the availability of Arabic datasets online, [41] suggests that some of the existing Arabic corpora are not dedicated for classification, because either there are no defined classes such as 1.5 billion words Arabic Corpus [42], or the existing classes are not well-defined. Therefore, the authors propose a new pre-processed and filtered corpus "NADA", composed from two existing corpora: OSAC and DAA. The authors used the DDC hierarchical number system that allows for each main category to be divided into ten sub-categories… and so on. "NADA" has 10 categories in total, with 13,066 documents. We believe that the size is small with respect to the proposed number of categories.

In addition, [43] investigates text classification using the SVM classifier on two datasets that differ in languages (English and Portuguese). It was found that the Portuguese dataset needs more powerful document representations, such as the use of word order and syntactical and/or semantic information.

Overall, it is clear that the performance of classification algorithms in Arabic text classification is greatly influenced by the quality of data source, feature representation techniques as the irrelevant and redundant features of data degrade the accuracy and performance of the classifier. This work is an extension of our work [44] on single-label classification.

3. DATASET

3.1 Single-label Dataset

We propose a newly collected dataset, consisting of 89,189 Arabic articles, tagged under 4 main categories [Sports, Business, Middle East and Technology]. Using the web-scraping framework, Scrapy, we collected data from 7 popular news portals (youm7.com, cnbcarabia.com, skynewsarabic.com, Arabic.rt.com, tech-wd.com, arabic.cnn.com and beinsports.com).

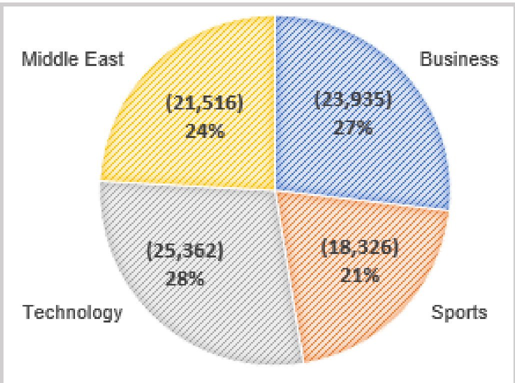


Figure 1. Single-labeled dataset distribution percentages.

With more than 32.5M words, all the articles in the dataset have no dialects and are written in Modern Standard Arabic (MSA). To avoid bias, it was essential to build a balanced corpus. On average, each category almost has 22k articles. Table 1 and Figure 1 show the exact distribution of the articles, with the count.

3.2 Multi-label Dataset

The same web scraping technique was used, in addition to others like BeautifulSoup and Selenium, in a hunt for websites that publish Arabic articles with several tags. Numerous amounts of multi-labeled articles, written in (MSA), were collected from 10 websites (cnbcarabia.com, beinsports.com, arabic.rt.com, tech-wd.com, youm7.com, aitnews.com, masrway.com, alarabiya.net, skynewsarabic.com and arabic.cnn.com).

Table 1. Articles count for each scraped news portal.

Websites	Classes	Articles Count
Sky News Arabia	Sports	7923
CNN Arabia	Sports	3800
	Tech.	1680
	Middle East	21516
	Business	3908
Bein Sports	Sports	6603
Tech-wd	Tech.	23682
Arabic RT	Business	896
Youm7	Business	14478
CNBC Arabia	Business	4653

As it is shown in Figure 2, the first collected dataset "Dataset_1" has 284,860 articles in total and was used to train and test the multi-labeling classifiers, giving the results apparent in Table 3 and Table 5. After studying the precision, recall and F1-score of each label, we decided to refine the dataset by removing redundant articles found in categories like: "Business" and by enriching the rest of them. The resulting dataset is called "Dataset_2" and contains 293,363 articles.

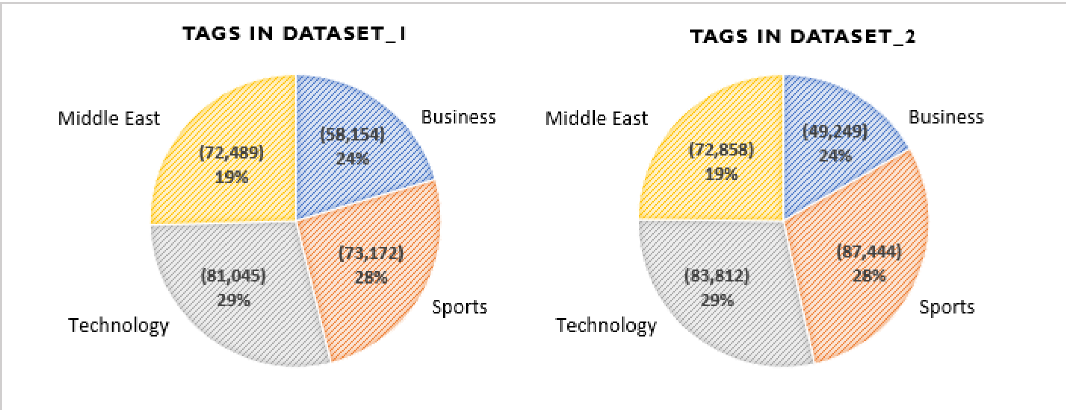


Figure 2. Article distribution in multi-labeled dataset.

4. Proposed Classification Systems

4.1 Text Features

Vectorization is the process of feature building, by turning text into numerical vectors. In text processing, words of the articles represent categorical features. This is a crucial step, as machine learning algorithms are unable to understand plain text. The most commonly used methods for this task are the Count Vectorizer and the Tf-IDF Vectorizer. Using a Count Vectorizer creates a Bag of words that counts the frequency of each word. However, using a Tf-IDF Vectorizer increases the value of the word proportionally to its count in a document, but is inversely proportional to its frequency in the corpus. The Tf-IDF Vectorizer is composed by two terms:

- Term Frequency (TF): measures how frequently a word occurs in an article. Since every article is different in length, it is possible that a term would appear much more times in long articles than in shorter ones.
- Inverse Document Frequency (IDF): measures how important a word is by weighing down the frequent terms and scaling up the rare ones.

It should be noted that unigram features are used in this work, as they reported better results when compared to bigram features. To show the effects of the using each of vectorizers on the model's accuracies, we made a comparison to rule out the best method. We used a portion of the dataset, containing approximately 40k articles belonging to 3 categories: (Middle East, Business and Sports). Figure 3 shows the results of the comparison, where the higher accuracy percentages were scored by the models using the Tf-IDF Vectorizer, which we decide to adopt in our work.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

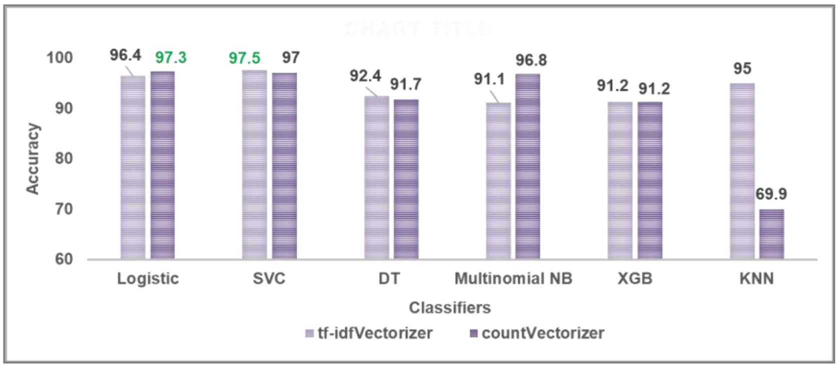


Figure 3. Accuracy comparison between TF-IDF Vectorizer and Count Vectorizer.

In addition to that, we put together our own custom-made stop words list and tested it against the NLTK built-in list. Higher accuracies were achieved using our list and we used it in further experiments.

Lastly, Figure 4 describes the overall workflow of our system. As this is a supervised machine learning approach, labels were one-hot encoded using sk-learn's Label Encoder and fed to the algorithms along side the vectors, during the training of the models.

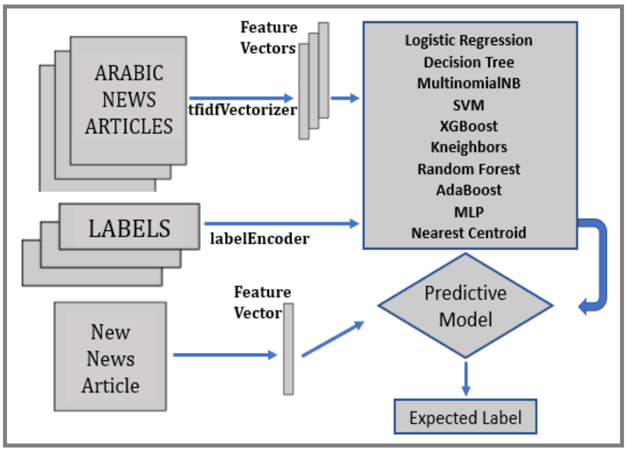


Figure 4. Summary of the work-flow of the classifiers.

4.2 Selected Classifiers

There exists an array of supervised shallow learning classifiers that are suitable to perform the text classification task. These classifiers have to map input data to a specific predicted category. We compared the results of 10 classifiers, in addition to a majority voting classifier. The classifiers are:

- Logistic Regression: Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.
- Multinomial Naïve Bayes: Using Bayes Theorem, this classifier calculates the probability of each label for a given data, then outputs the label with the highest probability. The classifier assumes that the attributes are independent of each other. In other words, the presence of one feature does not affect the presence of another; therefore, all the attributes contribute equally in producing the output.
- Decision Tree: This classifier resembles a tree, with each node representing a feature/attribute and each corresponding leaf representing a result. Each branch represents a condition and whenever a condition is answered, a new condition will be distributed recursively until a conclusion is reached. Recursion is used to partition the tree into a number of conditions with their outcomes.

- Support Vector Machines (SVM): This is a supervised non-probabilistic binary linear classifier that is extremely popular and robust. It constructs a model and outputs a line, known as the hyperplane, between classes. This hyperplane separates the data into classes. Both linear and nonlinear classification can be performed by the SVM classifier. The hyperplane can be written as the vector of input articles xx satisfying $w.x - b = 0w.x - b = 0$, where ww is the normal vector to the hyperplane and bb is the bias.
- Random Forest: This is a supervised ensemble learning-based classifier. It uses an array of decision trees. The outcome class is determined as an aggregate of such trees. Technically, given a set of articles $x_1, x_2, \dots, x_n x_1, x_2, \dots, x_n$ and their corresponding classes $y_1, y_2, \dots, y_n y_1, y_2, \dots, y_n$, each classification tree fb is trained using a random sample $(X_b, Y_b)(X_b, Y_b)$, where b ranges from 1 to the total number of trees. The predicted class shall be produced using a majority vote of all used trees.
- XGBoost Classifier: This is a supervised classifier, which has gained popularity because of winning a good number of Kaggle challenges. Like Random Forest, it is an ensemble technique of decision trees and a variant of gradient boosting algorithm.
- Multi-layer Perceptron (MLP): This is a supervised classifier. It consists of three (or more) layers of neuron nodes (an input and an output layer with one or more hidden layers). Each node of one layer is connected to the nodes of the next layer and uses a non-linear activation function to produce output.
- KNeighbors Classifier: This is a supervised classifier. In order to classify a given data point, we take into consideration the number of nearest neighbors of this point. Each neighbor votes for a class and the class with the highest vote is taken as the prediction. In other words, the major vote of the point's neighbors will determine the class of this point.
- Nearest Centroid Classifier: This is a supervised classifier. It's a no parameter algorithm, where each class is represented by the centroid of its members. It assigns to tested articles the label of the class of training samples with mean (centroid) closest to the article.
- AdaBoost Classifier: This is a supervised classifier. It is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset, but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.[3]
- Voting Classifier: It is a very interesting ensemble solution. It is not an actual classifier but a wrapper for a set of different classifiers. The final decision on a prediction is taken by majority vote.

It should be noted that only a number of supervised classical classifiers are suitable for the multi-labeling tasks and they are implemented using specific methods. For example, Logistic Regression and XGBoost both can classify text into multiple labels if each of them is wrapped in a OneVSRest Classifier. This will divide the bigger classification problem into many sub-problems.

We used the TF-IDF technique to vectorize the articles and used the default hyperparameters for each classifier. To encode the labels, we used MultiLabelBinarizer () that returns the string labels assigned to each article in a one-hot encoded format.

5. Experimental Results and Discussion

5.1 Setup and Pre-processing

5.1.1 Single-label Text Classification

The main objective is to conduct a comparative study on 11 classification models, testing them in classifying Arabic news categories. All classifiers are implemented using the popular scikit-learn (machine learning in Python). The experiment begins with classifying the articles, then analyzing the results and determining the best classifier. After that, the same algorithms will be trained and tested on another newly reported dataset called 'Akhbarona' [34]-[44], that divides the articles into 7 classes.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

We used the 80/20 ratio to split our dataset, where 80% of the data is in the training set, consisting of 71,707 articles and 20% in the testing set, containing 17,432 articles. More than 344k features were extracted from the training set. It should be noted that 10% of the training dataset is used for validation purposes in order to fine-tune parameters.

We report the accuracy score for each classifier, to evaluate their performance, which is calculated as the ratio of the number of correctly classified articles.

Moreover, cleaning and pre-processing the text are mandatory and recommended specially for text collected from the web. All non-Arabic content is removed and the articles are processed by eliminating punctuation, isolated characters, qur'anic symbols, elongation and other marks, along with the stop words.

Because the dataset is large enough to assign enough samples for each Arabic character, we believed that the normalization step is not necessary. In contrast to most research works on Arabic computational linguistics that apply normalization on the collected corpus, we skipped it as it can affect the meaning of some Arabic words. The stemming step is also skipped, as it results in a less deep view of the semantic relationships of the words, as it is concluded in [45].

5.1.2 Multi-label Text Classification

For this classifying approach, the dataset is split into 80% training set consisting of 118,700 labeled articles and 20% testing set consisting of 29,676 articles. The same text pre-processing steps used on the single-labeled dataset are used on the multi-labeling dataset.

A portion of the multi-labeled dataset proposed earlier, "Dataset_2" has been used to train and test the models. We chose to train on the labels with the highest frequency, because the performance of supervised deep learning classifiers is highly dependent on the number of instances for each label. Figure 5 shows the count of the 21 labels chosen from the dataset.

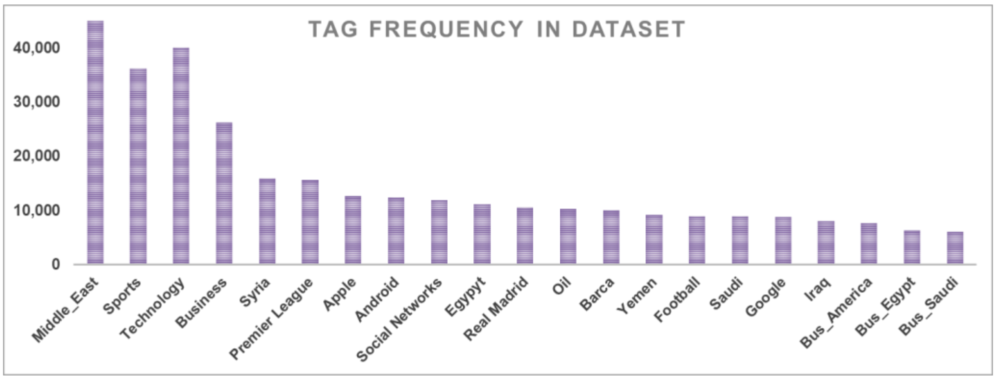


Figure 5. Tag count used in multi-labeling experiment using "Dataset-2".

One evaluating metric we used is a custom accuracy metric, to evaluate the accuracy of the predictions. It calculates the ratio of correctly predicted tags (output as 1) over total expected tags (originally 1 in dataset). The more correct labels the model predicts, the more accurate it is. We choose a threshold of 50%, meaning that if the probability percentage of the tag is equal to or higher than 50%, then its value will be set as 1.

The second metric is the hamming loss, which is a commonly used metric for multi-labeling tasks. It is the fraction of wrongly predicted labels to the total number of labels. The smaller the value, the better results the model is achieving. Figure 6 shows the relative distribution of the tags in "Dataset_2", where the highest number of tags for an article is 6 and the lowest is 2.

5.2 Performance Evaluation

5.2.1 Single-label Text Classification

All the classifier models were implemented using Scikit-learn and by using the default hyper-parameters as a black-box, with the addition of L1 penalty for some of the classifiers. The proposed classifiers were tested using the testing set. Figure 7 shows the accuracy scores of each classifier. The percentages are exceptionally high and prove the strength of the default hyper-parameters used in the system.

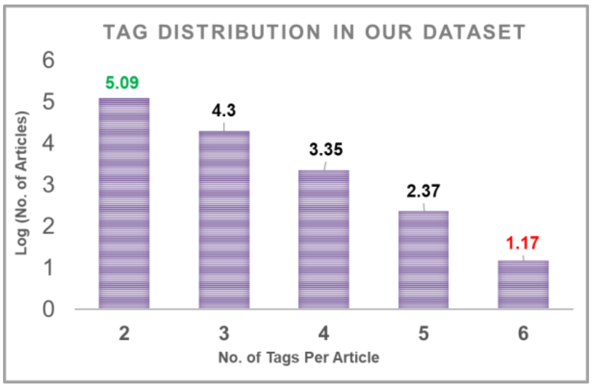


Figure 6. Tag count used in multi-labeling experiments.

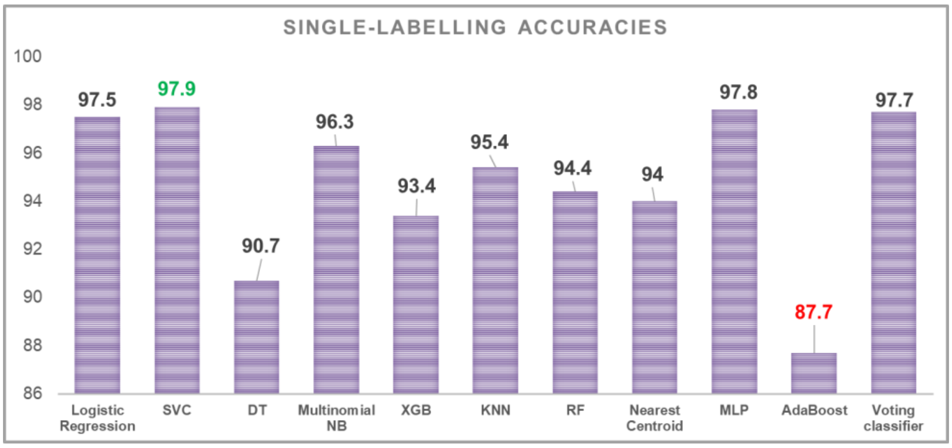


Figure 7. Accuracies for single-labeling classifiers using our dataset.

Producing the best result of 97.9% is the SVM classifier, while the worst percentage was produced by Ada-Boost. 4 classifiers achieved scores between 97.5% and 97.9% and the overall average of the percentages was 94.8%.

Two of the algorithms (KNeighbors and MultinomialNB) scored higher than the average, with percentages of 95.4% and 96.3%, respectively. The other models scored lower than the average with percentages ranging from 87.7% to 94.4%. The confusion matrix for SVM, the best classifier, is shown in Figure 8, while Figure 9 shows the matrix of the worst classifier Ada-Boost.

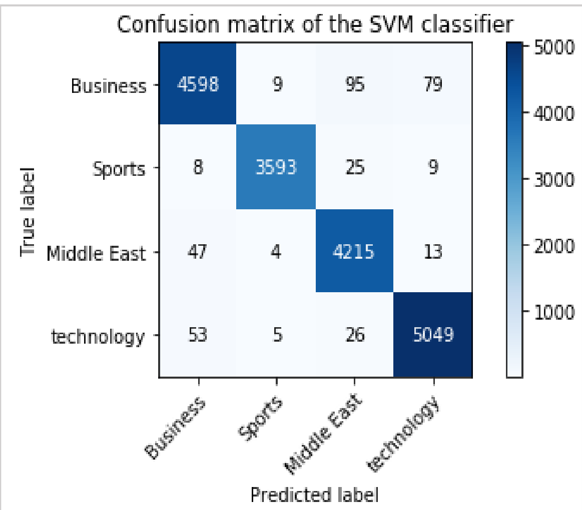


Figure 8. Confusion matrix for the best classifier.

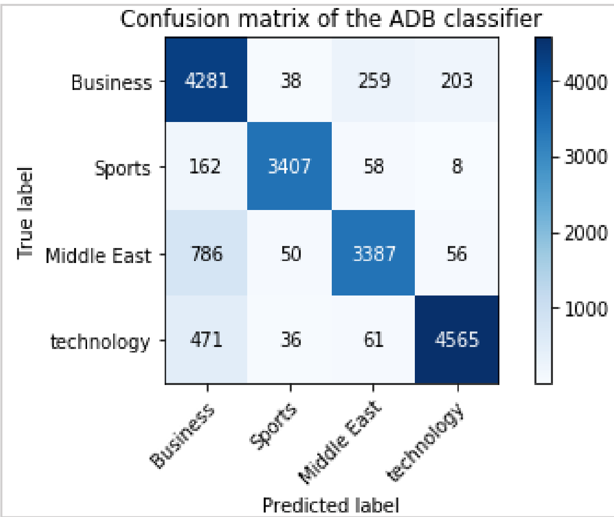


Figure 9. Confusion matrix for the worst classifier.

The matrices highlight that the 2 categories (Business and Middle East) had the greatest number of misclassifications and we believe that this is caused by feature similarity between the two. Table 2 shows the scores of the additional accuracy metrics used. The majority voting classifier and the SVM achieved the highest F1-score of 97.9%. The lowest score of 87.7% was produced by AdaBoost.

Table 2. Accuracy metrics for classifiers tested on our dataset.

Algorithms	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.98	0.98	0.98
SVC	0.98	0.98	0.98
DT Classifier	0.91	0.91	0.91
Multinomial NB	0.96	0.96	0.96
XGB Classifier	0.94	0.93	0.94
KNN Classifier	0.95	0.95	0.95
RF Classifier	0.95	0.94	0.94
Nearest Centroid	0.95	0.94	0.94
Ada-Boost Classifier	0.89	0.88	0.88
MLP Classifier	0.98	0.95	0.95
Voting Classifier	0.98	0.98	0.98

5.2.2 Multi-label Text Classification

Table 4 displays the evaluation metrics scores of both OVR-Logistic Regression and the OVR-XGBoost using “Dataste_2”. The average of the accuracies is 89.4%. XGBoost scored the highest of the two with a 91.3 % accuracy, while Logistic Regression scored an 87.6% accuracy. The hamming loss scores were low, where XGBoost scored the lowest with a percentage of 1.54% and Logistic Regression scored a percentage of 1.8%. The results prove that the XGBoost classifier was the better of the two.

Table 3. Evaluation metrics for multi-label classification using “Dataset_1”.

Evaluation metrics	**OVR-Logistic Regression**	**OVR - XGBoost**
Custom Accuracy	81.3%	84.7%
Hamming Loss	2.24%	2.22%

Table 4. Evaluation metrics for multi-label classification using “Dataset_2".

Evaluation metrics	**OVR-Logistic Regression**	**OVR - XGBoost**
Custom Accuracy	87.6%	91.3%
Hamming Loss	1.8%	1.54%

Table 5. Accuracy metrics for OVR-XGBoost classifier using “Dataset_1”.

Labels	**Precision**	**Recall**	**F1-score**	**Support**
Business	0.99	0.98	0.98	5154
Oil	0.92	0.98	0.91	2026
Business-America	0.91	0.72	0.81	1549
Business-Egypt	0.93	0.92	0.92	1197
Business-Saudi	0.86	0.83	0.85	1154
Middle East	1.00	1.00	1.00	9164
Syria	0.95	0.91	0.93	3181
Egypt	0.96	0.90	0.93	2270
Yemen	0.95	0.87	0.91	1774
Saudi	0.88	0.82	0.85	1759
Iraq	0.94	0.86	0.90	1616
Sports	1.00	0.99	0.99	7428
Premier League	0.92	0.91	0.92	3193
Real Madrid	0.90	0.90	0.90	2120

Barca	0.91	0.90	0.90	2048
Football	0.86	0.43	0.57	1856
Technology	0.98	0.98	0.98	6529
Android	0.62	0.25	0.36	2187
Apple	0.55	0.14	0.22	1566
Google	0.61	0.33	0.43	2319
Social Media	0.74	0.19	0.30	1630

Lastly, we calculated the F1-scores, precision and recall for each label predicted by the XGBoost classifier. The results shown in Table 6 slightly vary due to the imbalance in support numbers.

Table 6. Accuracy metrics for OVR-XGBoost classifier using "Dataset_2".

Labels	Precision	Recall	F1-score	Support
Business	0.97	0.94	0.95	5116
Oil	0.91	0.89	0.90	2041
Business-America	0.90	0.69	0.78	1557
Business-Egypt	0.91	0.84	0.87	1163
Business-Saudi	0.84	0.72	0.78	1162
Middle East	0.98	0.98	0.98	9197
Syria	0.94	0.90	0.92	3222
Egypt	0.93	0.86	0.90	2224
Yemen	0.94	0.87	0.90	1804
Saudi	0.83	0.76	0.79	1710
Iraq	0.92	0.86	0.89	1633
Sports	1.00	0.99	0.99	7309
Premier League	0.92	0.90	0.91	3136
Real Madrid	0.92	0.87	0.89	2169
Barca	0.90	0.88	0.89	2041
Football	0.81	0.43	0.56	1757
Technology	1.00	0.99	0.99	8054
Android	0.89	0.87	0.88	2493
Apple	0.93	0.86	0.89	2463
Google	0.87	0.87	0.87	1715
Social Media	0.94	0.90	0.92	2433

5.3 Sample Experiments

5.3.1 Single-label Text Classification

This phase is divided into 2 parts. The first part is to test the performance of the best classifier (SVM) by checking the predicted class of an article taken from the testing set.

Figure 10 shows an example of an article grabbed from the testing set and originally tagged as Technology. The SVM model assigned the same tag for the article with a confidence of 95.7%. For the record, our model has clearly shown how robust and coherent it is by showing a confidence of 99.6%.

Moreover, we further studied the predictions of the model on the test set by checking a portion of the misclassified articles. In our investigation, the model proved that some of the articles were a good fit under the predicted categories more than the originally assigned categories by the news website. In Figure 11, we show an article that is tagged under the "Technology" category. However, after checking the article, we are convinced that the "Business" category is more suited for this article which is the same category that was predicted by the SVM classifier. This proves again that the model is precise and trustworthy.

Finally, the article in Figure 12, taken from (cnbcarabia.com), was originally tagged as "Business". The SVM model was more biased for the "Middle East" tag with a probability percentage of 40.3%. The model was also giving a percentage of 37.7% for the "Business" tag. The small difference in the confidence was toward the more suited tag.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

مؤسس "أمازون" يتربع المنصب الثاني بين أثرى رجال العالم

تكنولوجيا نشر الأحد، 09 ابريل / نيسان 2017

نيويورك، الولايات المتحدة الامريكية (CNN)-- تنحّى جانباً يا وارين بافيت، فإن مؤسس شركة "أمازون"، جيف بيزوس، قد احتلّ منصبك كثاني أثرى رجل بالعالم.

إذ أظهرت إحصائية نشرها "مؤشر بولمبيرغ للمليارديرات" بأن ثروة بيزوس بلغت 75.6 مليار دولار، بينما قائمة المراتب ذاتها الصادرة من "فوربس" ذكرت بأن ثروة بيزوس تبلغ 75.2 مليار دولار "فقط".

وقد أضيف مليار دولار ونصف إلى ثروة بيزوس بعد كسب "أمازون" لتسعة مليارات دولار في سوق الأسهم خلال ساعات المداولة، الأربعاء.

قد يهمك أيضاً.. منهم مؤسس آبل السوري.. شركات لم تكن لتوجد دون المهاجرين

وقد حظيت الشركة بدعم قوي بأسهمها، بعد تصريح لأحد المحللين بمؤسسة "باركليز" بأن "أمازون" قال فيه إنه "من المرجّح أن تكون أول شركة برأس مال يبلغ التريليون دولار،" لتبلغ أسعار الأسهم معدلات قياسية بعدها.

01:01 أول طلب توصيل لأمازون بطائرة موجّهة يصل في 30 دقيقة فقط

وقد بلغت أسعار أسهم "أمازون" نمواً ملحوظاً بمقدار 16 في المائة مقارنة بالعام الماضي، وأبدى المستثمرون إعجابهم بشراء الشركة لموقع "سوق.كوم" المشهور في الشرق الأوسط والمتخصص بالتجارة الإلكترونية.

لكن ثروة بيزوس ليست محددة بشركة "أمازون" فحسب، فقد أسّس أيضاً شركة "Blue Origin" للأبحاث الفضائية، والتي تهدف إلى إنشاء بنية تحتية للسفر الفضائي مستقبلاً.

Figure 10. Example of a correctly classified news article.

حراك السودان | شرق أوسط | عالم | رياضة | تكنولوجيا | اقتصاد | منوعات | برامجنا

هواوي تطيح أبل وتتفوق على "الثلاثة الكبار"

3 مايو 2019 - 15:22 بتوقيت أبوظبي

سكاي نيوز عربية - أبوظبي

تخطت مبيعات عملاق الهواتف الصيني هواوي منافسه أبل ليحصل على المركز الثاني في مبيعات الهواتف العالمية في الربع الأول من العام خلف سامسونغ المتصدر.

وبحسب وكالة بلومبيرغ، تشير تقديرات شركة الأبحاث الدولية IDC إلى أن هواوي التي تعمل أيضا في تصنيع الشبكات العملاقة وتتجاهل مجموعة كبيرة من الاتهامات بالتجسس لصالح الصين، زادت من شحنات الهواتف بنسبة 50 في المائة عن العام السابق.

وهواوي هو الاسم الوحيد من بين أكبر أربع شركات في العالم يتمكن من توسيع أحجام المبيعات، في حين انخفض إجمالي السوق للربع السادس على التوالي.

وحققت هواوي مكاسب مطردة مقارنة بأبل وسامسونغ، مع الاعتماد على مجموعة متطورة من الأجهزة، لا سيما في سوقها بالصين.

وتوقع صانع آيفون هذا الأسبوع مبيعات ربع سنوية تفوق تقديرات المحللين، مما يشير إلى أن الطلب على أجهزة آيفون قد استقر بعد فترة أعياد مخيبة للآمال.

كما تمكنت سامسونغ من رفع مبيعاتها مقارنة بالربع الرابع بفضل إطلاق مجموعة هواتف غلاكسي S10 في فبراير.

وتشير شركة الأبحاث الدولية إلى أن لدى هواوي فرصة كبيرة للحفاظ على المركز الثاني خلال هذا العام.

Figure 11. Example of an incorrectly classified news article as “Business”.

For the second part of the testing, we experimented with a recently reported dataset (Akhbarona). It’s an unbalanced dataset that consists of seven categories [Medicine, Politics, Sports, Religion, Culture, Technology and Business] holding 46,900 articles. The dataset is cleaned by removing elongation, punctuation, Arabic digits, isolated chars, qur’anic symbols, Latin letters and other marks. We split the dataset into 80% training and 20% testing.

It can be stated that lower accuracies are to be expected by the models for 2 reasons; the increase in the number of categories will lead to a higher possibility of misclassifying an article and the unbalanced dataset may steer the classifier to be biased to a certain class.

Table 7 demonstrates the accuracy results that have been obtained on Akhbarona dataset. The SVM classifier proved to be the best classifier by producing an accuracy of 94.4% on the test set. In contrast, the Ada-Boost classifier produced no more than 87.7%. The average of the accuracies is 90% only. Furthermore, four classifiers were producing a close result to the best classifier with a range from 94.4% to 93.9%. The KNeighbors classifier performed above the average with an accuracy of 90.8%. The other six classifiers performed below the average with accuracy scores ranging from 77.9% to 88.4%. Figures 13 and 14 show the confusion matrix of the best classifier (SMV) and the worst classifier (Ada-Boost).


الرئيسية آخر الأخبار أخبار الشركات قطاعات برامج CNBC عربية فيديو CNBC عربية الأسواق

طباعة

بدء أول محادثات رسمية بين الكوريتين منذ أكثر من عامين

اتفقت كوريا الشمالية والجنوبية على اجراء محادثات عسكرية لازالة التوتر في شبه الجزيرة وذلك في اعقاب اللقاء الاول بين الجانبين منذ عامين كما أفادت وزارة الوحدة الكورية الجنوبية.

وجاء في بيان مشترك أن الجانبين اتفقا على "نزع فتيل التوتر العسكري الحالي واجراء محادثات عسكرية حول المسألة".

وقال متحدث باسم الوزارة للصحافيين أن المحادثات التي تجري في بانمونجوم في المنطقة المنزوعة السلاح بين شطري شبه الجزيرة بدأت بإعلان افتتاحي أدلى به رئيس الوفد الكوري الجنوبي الى هذه المحادثات.

هذا وأعلنت كوريا الشمالية أن وفد رفيع المستوى سيحضر الالعاب الاولمبية الشتوية في كوريا الجنوبية.

Figure 12. Another example of an article classified by SVM.

Table 7. Classifiers' accuracies on "Akbarona".

Algorithms	**Accuracy %**
Logistic Regression	93.9
SVC	**94.4**
DT Classifier	83.0
Multinomial NB	88.0
XGB Classifier	88.4
KNN Classifier	90.8
RF Classifier	87.8
Nearest Centroid	86.2
Ada-Boost Classifier	77.9
MLP Classifier	94.1
Voting Classifier	94.3

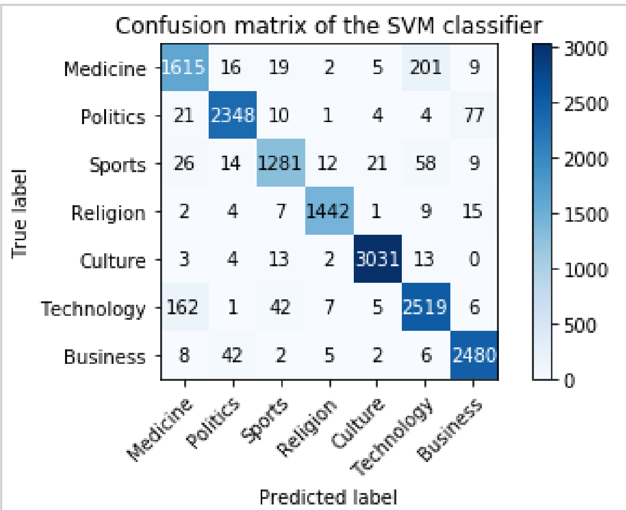


Figure 13. Confusion matrix for the best classifier on "Akhbarona".

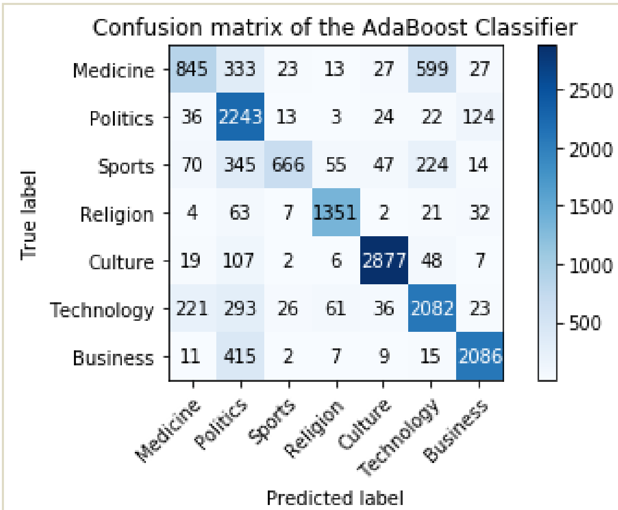


Figure 14. Confusion matrix for the worst classifier on "Akhbarona".

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

In Table 8, four different classifiers: (SVM, Logistic Regression, MLP and the voting classifier) scored the highest F1-score of 94%, while Ada-Boost scored the lowest score of 78%.

Table 8. Accuracy metrics for classifiers testing on “Akbarona”.

Algorithms	Precision	Recall	F1-score
Logistic Regression	0.94	0.94	0.94
SVC	0.94	0.94	0.94
DT Classifier	0.83	0.83	0.83
Multinomial NB	0.91	0.88	0.88
XGB Classifier	0.89	0.88	0.88
KNN Classifier	0.91	0.91	0.91
RF Classifier	0.88	0.88	0.88
Nearest Centroid	0.89	0.86	0.87
Ada-Boost Classifier	0.80	0.78	0.78
MLP Classifier	0.94	0.94	0.94
Voting Classifier	0.94	0.94	0.94

5.3.2 Multi-label Text Classification

During this phase, we test the performance of the OneVsRestXGBoost Classifier on recently published articles. Figure 15 shows an article taken from “Arabic.cnn.com”. The article discusses how the coronavirus could possibly impact the smartphones industry, since most of these factories are located in China. It’s originally tagged under “Business” only. Looking at the content of the article, it seems as it heavily talks about Technology as well. The XGBoost classifier picked up on both topics and predicted “Technology” and “Business” labels.

هل يؤثر فيروس كورونا على صناعة الهواتف الذكية؟

اقتصاد نشر الخميس، 06 فبراير / شباط 2020

سان فرانسيسكو، الولايات المتحدة الأمريكية (CNN) -- حذرت أكبر شركة مصنعة لشرائح الهواتف الذكية وأجهزة المودم في العالم من أن فيروس كورونا قد يعرقل صناعة الهواتف المحمولة حول العالم.

وقالت "كوالكوم" يوم الأربعاء إنها خفضت توجيه أرباحها للربع القادم جزئياً بسبب تفشي الفيروس في الصين. وقال المدير المالي للشركة، أكاش بالكيوالا، خلال محادثة الأرباح الأخيرة إنه "هناك شكوك كبيرة حول تأثير فيروس كورونا على الطلب على الهواتف المحمولة وسلسلة التوريد".

وقد تسبب فيروس كورونا في مقتل أكثر من 560 شخصاً وإصابة أكثر من 28 ألف شخص في جميع أنحاء العالم، معظمهم في الصين. وقد أثر تفشي الفيروس بشكل كبير على الأعمال التجارية العالمية، حيث أغلقت شركات التجزئة مئات المتاجر، وألغت شركات الطيران رحلاتها من وإلى الصين، وهبطت أسعار النفط الخام في وقت سابق من هذا الأسبوع.

ومن المحتمل أن تتأثر أيضاً صناعة الهواتف الذكية، التي تعتمد بشكل كبير على الصين في التصنيع والمبيعات. وتقول "كوالكوم" إن حوالي نصف إيرادات الشركة العام الماضي جاءت من الصين، وفقاً لتقريرها السنوي الأخير، كما أن العديد من عملائها، بما في ذلك شركة آبل، لديهم حضور رئيسي في الصين.

وقال ستيف مولينكوبف، الرئيس التنفيذي للشركة يوم الأربعاء إنه "مع استمرار تطور وضع فيروس كورونا، نفكر بالعديد من موظفي كوالكوم في الصين، وعملائنا وموردينا وعائلاتهم، وكذلك أولئك الذين تأثروا بهذا الوضع غير المسبوق".

وكانت قد أعلنت شركة صناعة الشرائح عن عوائد بلغت 5.08 مليار دولار للربع المنتهي في ديسمبر/ كانون الأول من العام 2019، بزيادة قدرها 5٪ عن ذات الفترة من العام الماضي، بينما انخفضت أرباحها بنسبة 13٪.

Figure 15. Example of a news article classified by OVR-XGBoost.

Another interesting article that has been correctly predicted by the model is shown in Figure 16. This time, it is taken from the “arabic.rt.com” website, where it was originally tagged with “Sports”, “Premier League”, “Real Madrid” and “Barcelona”. All of them were accurately predicted, without missing a single one. In Figure 17, the article originally is tagged under the “Business” category. Our trained classifier added to this tag two other labels: “Oil” and “Saudi Business”. This shows how specific the predictions by the model are.

سانشيز قد ينضم للريال في الميركاتو الشتوي المقبل

تاريخ النشر: 13.11.2018 | GMT 14:35 | الرياضة

A- A A+ | 737 | انسخ الرابط

أعجبني ٣,٢ ألف | مشاركة | تابعوا RT على | Google News

وبحسب الصحيفة، فاللاعب السابق لبرشلونة وآرسنال، غير مرتاح في ملعب "أولد ترافورد" معقل "الشياطين الحمر"، ويشعر بالملل، والوحده، والتجاهل.

وذكرت الصحيفة أن سانشيز على علم بأن ريال مدريد يرغب في التعاقد مع مهاجم خلال الميركاتو الشتوي المقبل، ولهذا طلب من وكيل أعماله فيرناندو فيلتسيفيتش نقله إلى القلعة البيضاء.

وأشارت الصحيفة إلى أن ما قد يعرقل إتمام هذه الصفقة، هو راتب سانشيز الضخم، فالمهاجم التشيلي هو الأعلى أجرا في الدوري الإنجليزي الممتاز لكرة القدم، ويتقاضى 23 مليون يورو سنويا.

وفي حال انضمام سانشيز للريال، فسيكون بمثابه هدية رأس السنة من فلورنتينو بيريز رئيس ريال مدريد لمدرب الفريق سانتياغو سولاري، الذي ثبت أمس بشكل رسمي على رأس الإدارة الفنية للميرينغي.

وسبق لسانشيز اللعب في الدوري الإسباني وبالتحديد في فريق برشلونة، بين العامين 2011 و2014.

Figure 16. Example of a correctly classified news article by OVR-XGBoost.

السعودية تدشن عصر الغاز بتطوير حقل "الجافورة" العملاق

قبل 14 ساعة

أعلنت السعودية، الجمعة، أن إنتاج حقل الجافورة من الغاز الطبيعي قد يصل في حال اكتمال تطويره إلى 2.2 تريليون قدم مكعب بحلول عام 2036، كما سيحقق دخلا صافيا بنحو 8.6 مليار دولار سنويا.

وعقدت اللجنة العليا للمواد الهيدروكربونية اجتماعا برئاسة ولي العهد السعودي الأمير محمد بن سلمان، اطلعت فيه على خطط تطوير الحقل العملاق في المنطقة الشرقية الذي يعد أكبر حقل للغاز غير المصاحب غير التقليدي يتم اكتشافه في السعودية.

وتم اكتشاف الحقل العملاق بطول 170 كيلومترا وعرض 100 كيلو متر، ويقدر حجم موارد الغاز في مكمنه بنحو 200 ترليون قدم مكعب من الغاز الرطب، الذي يحتوي على سوائل الغاز في الصناعات البتروكيماوية والمكثفات ذات القيمة العالية.

وأثنى الأمير محمد بن سلمان على جهود شركة أرامكو في تطوير الحقل بحجم استثمارات تصل إلى 110 مليارات دولار (412 مليار ريال)، وستؤدي مراحل تطويره إلى تزايد إنتاج الحقل من الغاز تدريجياً ليصل في حال اكتمال تطويره إلى 2.2 ترليون قدم مكعب عام 2036، وتمثل نحو 25 بالمائة من الإنتاج الحالي.

وبسبب خاصية الحقل سيكون قادراً على إنتاج نحو 130 ألف برميل يومياً من الإيثان وتمثل نحو 40 بالمائة من الإنتاج الحالي، ونحو 500 ألف برميل يومياً من سوائل الغاز والمكثفات اللازمة للصناعات البتروكيماوية وتمثل نحو 34 بالمائة من الإنتاج الحالي.

ووجه ولي العهد السعودي بأن تكون أولوية تخصيص إنتاج الحقل من الغاز وسوائله للقطاعات المحلية في الصناعة والكهرباء وتحلية المياه والتعدين وغيرها لمواكبة معدلات النمو الطموحة وفق رؤية 2030، وفق ما ذكرت وكالة الأنباء السعودية (واس).

وأشار إلى أن تطوير الحقل سيحقق طوال 22 عاما من بداية تطويره دخلا صافياً للحكومة بنحو 8.6 مليار دولار سنوياً (32 مليار ريال)، ويرفد الناتج المحلي الإجمالي بما يقدر بـ20 مليار دولار (75 مليار ريال) سنوياً.

Figure 17. Example of a news article classified by OVR-XGBoost.

الحوثي يهاجم دول الخليج ويتهم الإخوان المسلمين باستقدام تنظيم القاعدة والقتال معه باليمن

الشرق الأوسط | نشر الأربعاء، 11 مارس / آذار 2015

وقال الحوثي، في خطاب جديد له أمام أنصاره مساء الثلاثاء بمناسبة "يوم الشهيد" ألقاه عبر شاشات التلفزة، إن قوى خارجية ومحلية وفي مقدمتها حزب الإصلاح "استقدموا التكفيريين ووفروا لهم السلاح والغطاء السياسي والإعلامي ومدوهم بكل وسائل القوة للهيمنة على هذا البلد والفتك به والقضاء عليه بكل الوسائل البشعة ومن ثم احتلاله باسم مكافحة الإرهاب والقضاء عليه."

واتهم الحوثي حزب الإصلاح بـ"العمل ضد الثورة الشعبية والمسيرة القرآنية" على حد وصفه مضيفا: "نحن ندرك وبات من الحقائق ان حزب الإصلاح - الذي يبرز في مواقف واضحة ضد الثورة الشعبية والمسيرة القرآنية – متواطئ مع القاعدة وينسق معه في عمليات قتالية."

وتوجه الحوثي، الذي يحظى بدعم كبير من إيران، إلى القوى الغربية بالقول إنها تراهن على "الخاسرين" وانتقد دول الخليج ومواقفها من سيطرته على مفاصل الدولة بصنعاء قائلا: "هل هناك من موقف عادل ومنصف لدول مجلس التعاون الخليجي في مصلحة الشعب اليمني؟ هل من موقف لهذه الدول سوى ارسال الدعم بالمال والسلاح للعناصر التكفيرية ومن تهيئة المناخ للقاعدة في المحافظات الجنوبية؟"

ورأى الحوثي في خطابه أن ما وصفها بـ"القوى التكفيرية" تحاول السيطرة على معسكرات الجيش وتعلن انضمامها إلى داعش في العراق وسوريا.

Figure 18. Example of a misclassified news article by OVR-XGBoost from the test set.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

Lastly, in Figure 18, we show an example of a misclassified article grabbed form the testing set. The article is originally tagged as “Middle East”, “Egypt” and “Yemen”. The predicted results show that the model has agreed with the author to a certain level, by predicting “Middle East”, “Yemen”, in addition to two other additional tags: “Iraq” and “Syria”. Along with the absence of the tag “Egypt”, we found that the model’s overall prediction was much more suitable for the article and its content, proving that it can rectify human errors.

6. CONCLUSIONS

In summary, this paper has presented both a multi-class text classifier system for Arabic news articles and a multi-label classifying system. We propose a single-labeled dataset that holds over 89k Arabic news articles divided into 4 categories, from seven websites. Another dataset is also proposed that contains 293k multi-labeled Arabic articles along with their tags, scraped from 10 different websites. The first dataset was examined by 11 different classifiers, all trained and tested using the single-labeled dataset. From 87% to 97% is the range of final accuracies, where the SVM classifier scored the top accuracy and F1-score. The voting classifier was used in hopes of improving the accuracy, but the resulting percentage is comparable to the SVM classifier’s score. To further explore how robust our proposed system is, we conducted additional experiments on a recently reported dataset “Akhbarona”. The dataset has 7 classes and the results of using it for training and testing the same classifiers were as good as on our dataset. The highest accuracy was scored by the SVM classifier as well. The second dataset was examined by implementing and comparing the results of two different classifiers. A custom accuracy metric was implemented to evaluate the performance along with hamming loss metric. The OVR-XGBoost classifier performed better than OVR-Logistic Regression classifier, scoring 91.3% accuracy, while Logistic achieved 87.6%.

In future, we intend to increase the number of classes in the single-labeled dataset and the number of labels in the multi-labeled dataset, which will require more scraping scripts. We would also like to compare and study our results with some deep learning methods.

ACKNOWLEDGEMENT

We would like to thank the University of Sharjah for funding this work under targeted research project no.: 1702141151-P.

REFERENCES

[1] A. Elnagar and O. Einea, "BRAD 1.0: Book Reviews in Arabic dataset," Proc. of the IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1-8, DOI: 10.1109/AICCSA.2016.7945800, Agadir, Morocco, 2016.

[2] A. Elnagar, Y. Khalifa and A. Einea, "Hotel Arabic-reviews Dataset Construction for Sentiment Analysis Applications," Book Chapter in Intelligent Natural Language Processing: Trends and Applications, pp. 35-52, DOI: 10.1007/978-3-319-67056-0_3, 2017.

[3] A. Elnagar, L. Lulu and O. Einea, "An Annotated Huge Dataset for Standard and Colloquial Arabic Reviews for Subjective Sentiment Analysis," Procedia Computer Science, vol. 142, pp. 182-189, 2018.

[4] N. Boudad, R. Faizi, R. O. Thami and R. Chiheb, "Sentiment Analysis in Arabic: A Review of the Literature," Ain Shams Engineering Journal, vol. 9, pp. 2479-2490, 2017.

[5] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud and P. Duan, "Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification," Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING), pp. 2418–2427, Osaka, Japan, 2016.

[6] H. Almuaidi, S. Alqrainy and A. Ayesh, "Automated Tagging System and Tagset Design for Arabic Text," International Journal of Computational Linguistics Research, vol. 1, pp. 55-62, 2010.

[7] A. Al-Alwani and M. Beseiso, "Arabic Spam Filtering Using Bayesian Model," International Journal of Computer Applications, vol. 79, pp. 11-14, 2013.

[8] Y. Li, X. Nie and R. Huang, "Web Spam Classification Method Based on Deep Belief Networks," Expert Syst. Appl., vol. 96, pp. 261-270, 2018.

[9] S. Malmasi and M. Dras, "Language Identification Using Classifier Ensembles," Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, Association for Computational Linguistics, pp. 35–43, Hissar, Bulgaria, 2015.

[10] M. El-Haj, P. Rayson and M. Aboelezz, "Arabic Dialect Identification in the Context of Bivalency and Code-Switching," Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), ELRA, pp. 3622-3627, Miyazaki, Japan, 2018.

[11] N. Y. Habash, Introduction to Arabic Natural Language Processing, Synthesis Lectures on Human Language Technologies, Edited by Graeme Hirst, [Online], Available: https://doi.org/10.2200/S00277ED1V01Y201008HLT010, 2010.

[12] S. C. Dharmadhikari, M. Ingle and P. Kulkarni, "Empirical Studies on Machine Learning Based Text Classification Algorithms," Advanced Computing: An International Journal, vol. 2, pp. 161-169, 2011.

[13] C. C. Aggarwal and C. Zhai, "A Survey of Text Classification Algorithms," Mining Text Data, pp. 163-222, 2012.

[14] V. Korde and C. N. Mahender, "Text Classification and Classifiers: A Survey," International Journal of Artificial Intelligence & Applications, vol. 3, pp. 85-99, 2012.

[15] I. Hmeidi, M. Al-Ayyoub, N. A. Abdulla, A. A. Almodawar, R. Abooraig and N. A. Mahyoub, "Automatic Arabic Text Categorization: A Comprehensive Comparative Study," Journal of Information Science, vol. 41, no. 1, pp. 114-124, 2015.

[16] A. M. Sbou, "A Survey of Arabic Text Classification Models," International Journal of Informatics and Communication Technology, vol. 8, pp. 25-28, 2019.

[17] M. Saad and W. Ashour, "Arabic Text Classification Using Decision Tree," Proc. of the 12th International Workshop on Computer Science and Information Technologies (CSIT'2010), vol. 2, pp. 75-79, Moscow, Russia, 2010.

[18] F. Harrag, E. El-Qawasmeh and P. Pichappan, "Improving Arabic Text Categorization Using Decision Trees," Proc. of the 1st International Conference on Networked Digital Technologies, pp. 110-115, Ostrava, Czech Republic, 2009.

[19] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed and A. Alrajeh, "Automatic Arabic Text Classification," JADT 2008: 9es Journées Internationales d'Analyse Statistique des Données Textuelles, pp. 77-83, 2008.

[20] M. E. Kourdi, A. Bensaid and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," Workshop on Computational Approaches to Arabic Script-based Languages, DOI: 10.3115/1621804.1621819, 2004.

[21] H. M. Noaman, S. Elmougy, A. Ghoneim and T. T. Hamza, "Naive Bayes Classifier-based Arabic Document Categorization," Proc. of the 7th International Conference on Informatics and Systems (INFOS), pp. 1-5, Cairo, Egypt, 2010.

[22] S. Alsaleem, "Automated Arabic Text Categorization Using SVM and NB," International Arab Journal of e-Technology, vol. 2, no. 2, pp. 124-128, 2011.

[23] M. J. Bawaneh, M. Alkoffash and A. I. Rabea, "Arabic Text Classification Using K-NN and Naive Bayes," Journal of Computer Science, vol. 4, no. 7, pp. 600-605, 2008.

[24] T. F. Gharib, M. B. Habib and Z. T. Fayed, "Arabic Text Classification Using Support Vector Machines," International Journal of Computers and Their Applications, vol. 16, no. 4, pp. 192-199, 2009.

[25] F. Harrag and E. Al-Qawasmah, "Improving Arabic Text Categorization Using Neural Network with SVD," Journal of Digital Information Management, vol. 8, no. 4, pp. 233-239, 2010.

[26] I. Hmeidi, B. Hawashin and E. El-Qawasmeh, "Performance of KNN and SVM Classifiers on Full Word Arabic Articles," Advance Engineering Informatics, vol. 22, no. 1, pp. 106-111, 2008.

[27] S. Boukil, M. Biniz, F. E. Adnani, L. Cherrat and A. E. Moutaouakkil, "Arabic Text Classification Using Deep Learning Techniques," International Journal of Grid and Distributed Computing, vol. 11, pp. 103-114, 2018.

[28] F. A. Zaghoul and S. Al-Dhaheri, "Arabic Text Classification Based on Features Reduction Using Artificial Neural Networks," Proc. of the 15th International Conference on Computer Modelling and Simulation (UKSim), pp. 485-490, Cambridge, UK, 2013.

[29] M. M. Al-Tahrawi and S. N. Al-Khatib, "Arabic Text Classification Using Polynomial Networks," Journal of King Saud University- Computer and Inform. Sciences Archive, vol. 27, pp. 437-449, 2015.

[30] L. Lulu and A. Elnagar, "Automatic Arabic Dialect Classification Using Deep Learning Models," Procedia Computer Science, vol. 142, pp. 262-269, 2018.

[31] A. A. Altowayan and A. Elnagar, "Improving Arabic Sentiment Analysis with Sentiment-specific Embeddings," Proc. of IEEE International Conference on Big Data (Big Data'2017), pp. 4314-4320, Boston, MA, USA, 2017.

[32] A. Elnagar, R. Ismail, B. Alattas and A. Alfalasi, "Automatic Classification of Reciters of Quranic Audio Clips," Proc. of the 15th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1-6, Aqaba, Jordan, 2018.

[33] A. Elnagar and M. Lataifeh, "Predicting Quranic Audio Clips Reciters Using Classical Machine Learning Algorithms: A Comparative Study. In book: Recent Advances in NLP: The Case of Arabic Language, vol. 874, pp. 187-209, DOI: 10.1007/978-3-030-34614-0_10, 2020.

[34] A. Elnagar, O. Einea and R. A. Debsi, "Automatic Text Tagging of Arabic News Articles Using Ensemble Deep Learning Models," Proceedings of the 3rd International Conference on Natural Language and Speech Processing (ICNLSP), pp. 59-66, Trento, Italy, 2019.

[35] A. Elnagar, R. Al-Debsi and O. Einea, "Arabic Text Classification Using Deep Learning Models," Information Processing and Management, vol. 57, no. 1, pp. 102-121, 2020.

[36] A. El-Halees, "A Comparative Study on Arabic Text Classification," Egyptian Computer Science Journal, vol. 30, [Online], Available: http://ecsjournal.org/JournalArticle.aspx?articleID=193, 2008.

[37] R. Al-Shalabi and R. Obeidat, "Improving KNN Arabic Text Classification with N-Grams Based Document Indexing," Proc. of the 6th International Conference on Informatics and Systems, pp. 108-112, Cairo, Egypt, 2008.

[38] G. I. Raho, R. Al-Shalabi, G. Kanaan and A. Nassar, "Different Classification Algorithms Based on Arabic Text Classification: Feature Selection Comparative Study," International Journal of Advanced Computer Science and Applications, vol. 6, no. 2, pp. 192-195, 2015.

[39] A. Mesleh, "Chi Square Feature Extraction Based SVMs Arabic Language Text Categorization System," Journal of Computer Science, vol. 3, no. 6, pp. 430-435, 2007.

[40] B. Hawashin, A. Mansour and S. A. Aljawarneh, "An Efficient Feature Selection Method for Arabic Text Classification," Int. Journal of Computer Applications, vol. 83, no. 17, pp. 1-6, 0975-8887, 2013.

[41] N. Alalyani and S. L. Marie-Sainte, "NADA: New Arabic Dataset for Text Classification," International J. of Advanced Comp. Science and Applications, vol. 9, DOI: 10.14569/IJACSA.2018.090928, 2018.

[42] I. A. El-Khair, 1.5 Billion Words Arabic Corpus, Computer Science, Computation and Language, ARXIV, ABS/1611.04033, 2016.

[43] T. Gonçalves and P. Quaresma, "The Impact of NLP Techniques in the Multi-label Text Classification Problem," Intelligent Information Systems, vol. 25, pp. 424-428, 2004.

[44] L. A. Qadi, H. E. Rifai, S. Obaid and A. Elnagar, "Arabic Text Classification of News Articles Using Classical Supervised Classifiers," Proc. of the 2nd International Conference on New Trends in Computing Sciences (ICTCS), pp. 1-6, DOI: 10.1109/ICTCS.2019.8923073, Amman, Jordan, 2019.

[45] A. M. Hassanein and M. Nour, "A Proposed Model of Selecting Features for Classifying Arabic Text," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 275-290, 2019.

ملخص البحث:

يهدف هذا العمل الى وسم الوثائق بما يلائمها من أوصافٍ بناءً على خصائص المفردات المستخدمة فيها. ولتحقيق هذا الهدف، تم تكوين مجموعتي بيانات كبيرتين من المنافذ الخاصة بالأخبار باللغة العربية. تتكون مجموعة البيانات الأولى من مقالاتٍ موسومةٍ بوصفٍ واحدٍ من بين 4 حقول هي: (الأعمال، والشرق الأوسط، والتكنولوجيا، والرياضة). والثانية، فهي أكبر من الأولى، وتحتوي على مقالاتٍ موسومة بعدة أوصافٍ. وستُتاح مجموعتا البيانات لمجتمع البحث فيما يتعلق باللغويات الحاسوبية باللغة العربية.

ولفحص مدى الفائدة من مجموعتي البيانات، تم تطبيق مصفوفةٍ من عشرة مصنِّفات ضحلة التعلُّم. بالإضافة الى ذلك، تم تطبيق نموذج تجميع من أجل جمع أفضل المصنِّفات معاً في مصنِّفٍ واحدٍ اعتماداً على تصويت الأغلبية. وتراوح أداء المصنِّفات على مجموعة البيانات الأولى بين 87.7% لمصنِّف (Ada-Boost) و97.9% لمصنِّف (SVM).

وقد أكدّ تحليل بعض المقالات التي فشل النموذج في تصنيفها بشكل صحيح الحاجة الى تصنيفٍ متعدد الأوصاف على العكس من التصنيف القائم على وصفٍ واحد؛ للحصول على نتائج تصنيف أفضل. وقد جرى استخدام مصنِّفاتٍ متوافقة مع مهمات التصنيف متعدد الأوصاف؛ مثل (LR) و (XGBoost)، وتم اختبار هذه المصنِّفات على مجموعة البيانات الثانية. وتم تطوير مقياس للدقة لتقييم الأداء، الى جانب مقياس لفقْد الطرق. وأثبت المصنِّف (XGBoost) أنه الأفضل؛ إذ سجّل دقة بلغت 91.3%، في حين كانت دقة المصنِّف (LR) 87.6%.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

ARABIC SIGN LANGUAGE CHARACTERS RECOGNITION BASED ON A DEEP LEARNING APPROACH AND A SIMPLE LINEAR CLASSIFIER

Ahmad Hasasneh

(Received: 27-Apr.-2020, Revised: 22-Jun.-2020, Accepted: 18-Jul.-2020)

ABSTRACT

One of the best ways of communication between deaf people and hearing people is based on sign language or so-called hand gestures. In the Arab society, only deaf people and specialists could deal with Arabic sign language, which makes the deaf community narrow and thus communicating with normal people difficult. In addition to that, studying the problem of Arabic sign language recognition (ArSLR) has been paid attention recently, which emphasizes the necessity of investigating other approaches for such a problem. This paper proposes a novel ArSLR scheme based on an unsupervised deep learning algorithm, a deep belief network (DBN) coupled with a direct use of tiny images, which has been used to recognize and classify Arabic alphabetical letters. The use of deep learning contributed to extracting the most important features that are sparsely represented and played an important role in simplifying the overall recognition task. In total, around 6,000 samples of the 28 Arabic alphabetic signs have been used after resizing and normalization for feature extraction. The classification process was investigated using a softmax regression and achieved an overall accuracy of 83.32%, showing high reliability of the DBN-based Arabic alphabetical character recognition model. This model also achieved a sensitivity and a specificity of 70.5% and 96.2%, respectively.

KEYWORDS

Arabic sign language, Sign language recognition, Deep belief network, Softmax regression, Classification.

1. INTRODUCTION

The most natural ways that human beings used to communicate with each other are by using voice, gestures and human-machine interfaces. The last method is still very primitive and forces us to adapt to the machine requirements. Also, the use of voice signals to communicate with hearing-impaired people is impossible or not desirable at all. However, deaf signs or gesture signs of sign language can be desirable and used to communicate with deaf people during their daily life. It's well-known that sign language is the language of deaf people that depends on the body movements or any visual-manual way, particularly the human hands and arms, to convey meanings, where the deaf sign language is typically different from one language to another and from one country to another.

In the Arab society, the community of Arab deaf people is small and very limited; this is due to the fact that only specialists deal with them. Statistics show that over 3% of the Palestinian population are hearing-impaired [1]. Also, according to the Palestinian Central Bureau of Statistics, 19% of disabled Palestinian people are deaf and mute [2]. Indeed, helping those people is very important and thus developing technical systems capable of translating sign languages into text or spoken language is highly needed. Developing such systems will definitely participate in facilitating the communication between the hearing-impaired and hearing people. In addition to that, it has been shown that Arabic sign language (ArSL) is the most difficult recognition task among other foreign sign languages due to its unique structure and complex grammar [3], where the researchers in the Middle East and Arab countries started to pay attention to this problem in the early 1990s. Therefore, developing recognition systems for ArSL is still an open question and a challenging task, which involves three phases of recognition: the first phase is alphabets recognition; the second one is the recognition of an isolated word with one sign and finally, the recognition of a word that contains continuous signs. However, we have seen that most of the existing ArSLR models have been satisfactory for Arabic alphabet recognition with excellent accuracy. Figure 1 shows the 30 letters of Arabic alphabet.

Generally, ArSLR can be performed through two main phases: detection and classification. In the detection phase, each captured image is pre-processed, improved and then the Regions of Interest

A. Hasasneh is with Faculty of Eng. and IT, Palestine Ahliya University, Bethlehem, Palestine. Email: ahasasneh@paluniv.edu.ps

(ROI) are identified using a segmentation algorithm. The output of the segmentation process can thus be used to perform the classification process. Indeed, the accuracy and speed of detection play an important role in obtaining accurate and fast recognition process. In the recognition phase, a set of features are extracted from each segmented hand sign and then used to perform the recognition process. These features can, therefore, be used as a reference to understand the differences among the different signs.

Figure 1. **Left:** Original Arabic sign language alphabet [4]. **Right:** The corresponding 32x32 small images.

As mentioned earlier, ArSLR systems have only been paid attention recently [5]–[7]. Some of these attempts are vision-based ArSLR and used the K-nearest neighbor rule [5] or used the hidden Markov model (HMM) [6] and achieved promising results. Some others are sensor-based ArSLR and used the CyberGlove coupled with principal component analysis for feature extraction, followed by a support vector machine (SVM) [7]. Therefore, investigating and developing a new ArSLR model are important and using alternative approaches has to be considered. This paper thus proposes an alternative simpler Arabic sign recognition system based on deep feature extraction methods followed by a simple linear classifier method. Deep learning models have recently shown significant successes in different applications, such as robotics [8], neuroscience [9], traffic sign recognition [10], object detection and recognition [11], audio recognition [12], Bib number detection and recognition [13], Arabic handwritten recognition [10], [14] and image compression and information retrieval [15]-[16].

The rest of the paper is organized as follows. Section 2 presents an overview of the related works. Section 3 presents the proposed model for the recognition of 28 Arabic machine-print characters. Section 4 details the experimental results. Finally, the discussion and conclusions are presented in section 5.

2. RELATED WORKS

Generally, sign language recognition systems for American, British, Indian, Chinese, Turkish and many international sign languages have received much attention compared to the Arabic sign language. A review of the recent development in sign language recognition for foreign languages can be found in [17]–[20]. Also, most of the proposed approaches to the problem of ArSLR have given rise to sensor-based techniques [7], [21]-[22]. Sensor-based model usually employs sensors attached to the hand glove and a look-up table software is usually provided with the glove to be used for hand gesture recognition. However, image-based ArSLR techniques have recently emerged and have been investigated [23]–[29]. The task of image-based ArSLR typically requires firstly producing an appropriate code for the initial data and secondly using this code to classify and learn the alphabet in a fast and accurate manner. Image-based model usually uses video cameras to capture the movements of the hand. However, image-based techniques exhibit a number of challenges, including lighting conditions, image background, face and hand segmentation and different types of noise.

Different classification approaches (generative & discriminative methods) have recently been developed and used to address the problem of ArSLR and most of them focused on recognizing the signs of Arabic alphabet [23], [26]-[27], [30]–[33]. In particular, the authors in [23] developed a neuro-fuzzy system, which includes five main stages, including image acquisition, filtering, segmentation, hand outline detection followed by feature extraction. The conducted experiment considered the use of the bare hand and achieved a hit rate of 93.6%. The author in [32] has also introduced an automatic recognition of the Arabic sign language letters. For feature extraction, Hu's moments are used and for the classification process, the moment invariants are fed to an SVM. A correct classification rate of 87% was achieved.

The authors in [26] proposed an automatic ArSLR system, that translates isolated Arabic word signs into text, which involves four main stages: hand segmentation, tracking, feature extraction and classification. This proposed model achieved a correct recognition rate of 97% in signer-independent mode. Another recent ArSLR model based on optical flow-based features and HMM was proposed in [30]. In this model, the signs were transformed using four transformation techniques, including: Fourier Transform (MFT), Local Binary Pattern, Histogram of Oriented Gradients (HOG) and combination of HOG and Histogram of Optical Flow. The best classification result was achieved using HMM with MFT features with an accuracy of 99.11%.

Furthermore, the authors in [31] proposed an isolated sign language recognition system that extracts geometric features from a camera for the hand gesture and builds a geometric model for the hand gesture. Using the extracted geometric features, the recognition process of a specific gesture was then performed using the rule-based classifier. The proposed model was tested on seven Arabic words and achieved an overall classification rate of 95.3%. Another work in [32] described two dynamic sign language recognition systems based on two different methods; real-time (online) and offline ones. The comparison was made between the two methods and it was found that the recognition rate for the online hand gesture recognition is lower than the recognition rate for the offline system, which underlines that further enhancements are still needed to improve the real-time recognition performance.

Moreover, several recent attempts have been proposed to use recurrent neural networks (RNNs) [34], convolutional neural networks (CNNs) [27], [31] to achieve ArSLR. More specifically, the authors in [34] proposed to use RNNs together with colored gloves in their experiments. This proposed model achieved an accuracy rate of 89.7%, while a fully recurrent network improved the accuracy to 95.1%; however, it complicated the learning process. The authors in [31] proposed to use 3D CNNs to recognize 25 gestures of Arabic sign dictionary. In this model, the features were extracted with deep behaviour and the proposed model achieved a correct classification rate of 85% for the testing data. Finally, a similar ArSLR approach was proposed in [27] based on CNNs and direct use of softmax regression. When 50% of the dataset was used for training, this similar approach achieved 83.27% of correct classification rate, while when the training dataset was increased to 80%, the correct classification rate increased to 90.02%.

Although most of the current approaches have achieved excellent classification results, some of them are either based on sophisticated classifiers, such as [32], or based on complex learning methods, such as [26]-[27], [31], [33]. In contrast, this paper proposes a deep learning approach that uses a fast learning technique, a Restricted Boltzmann Machine (RBM) and a simple linear classification method. The simplification of the overall classification process of Arabic sign letters must also achieve accurate results. This hypothesis is based on improving the linear separation between signs of Arabic alphabet in the learning phase. This can be achieved by using a powerful machine learning method capable of extracting appropriate features that can be used later to generate an appropriate code for the classification phase. The proposed model is illustrated in the following section and has been recently proposed as a perspective study in our previous publication [35], where it is mainly based on Deep Belief Networks (DBNs) and softmax regression.

3. MATERIALS AND PROPOSED MODEL

3.1 ArSL Dataset and Data Preprocessing

The ArSL dataset used in this paper to test the proposed model was collected by Suez Canal University and used by [4]. In brief, this dataset consists of 210 gray-scale images representing the gestures of 30 Arabic letters; i.e., 7 images for each letter gesture. As stated in [4], the dataset has been captured with different rotations, under different illumination conditions and based on various volunteers who have different hand sizes. In our experiments, we used the first 28 signs of the Arabic letters, as shown in Figure 1. 50% of these images were repeated to create a sufficient dataset of a total of 6000 images, which were randomly distributed into groups to make them appropriate for epoch training method. The other sub-set was used for testing the proposed model.

One can see that ArSL images contain a lot of hand features, borders, corners and edges which will contribute and foster the learning network to extract localized and sparse features based on a deep learning approach. It has been shown that the typical input layer of DBN training should be approximately around 1000 pixels, which requires a significant reduction of the original images [8]. Therefore, all images were cropped and significantly reduced to 32×32 = 1024 pixels with a fixed scale, as shown in Figure 1 (right). Despite the big reduction, one can see that the reduced images remain fully recognizable. It has also been shown that DBNs are still capable of extracting interesting features from tiny images [8], [36]. To ensure the network to learn sparse and localized features with higher-order statistics, the tiny images were locally normalized with zero-mean and unit variance. It has recently been shown that the local normalization method achieved better results in terms of feature extraction and classification [8]. Consequently, the normalized tiny images have been used as the input vector to train the network and build the model.

3.2 Deep Learning Model

We developed a novel deep learning approach specifically for the classification of ArSL. The general workflow of the proposed model includes three main stages (see Figure 2), which can be summarized as follows: 1) image pre-processing, 2) unsupervised feature space construction and finally 3) Arabic sign language recognition. The first two steps of the proposed model have recently been carried out and published [35].

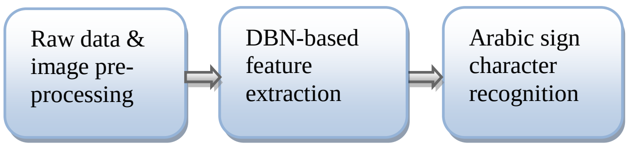


Figure 2. General workflow of the DBN-based model for Arabic sign character recognition using the normalized tiny images.

DBNs are probabilistic generative models composed of multiple RBM layers of latent stochastic variables [8], as illustrated in Figure 3 (right). The RBM is a powerful machine learning algorithm, which can be used to train deep networks in a greedy layer-wise way. The RBM is a bipartite undirected graphical model and consists of two layers (visible (v_i) and hidden (h_j) layers). The two layers are fully connected through a set of weights (w_{ij}) and biases $\{b_i, c_j\}$. Moreover, there is no connection between units of the same layer.

Typically, the input layer of RBM corresponds to the input normalized data. Thus, given the visible vector, the activation probability of the hidden layer can be computed as follows:

$$P\left(h_j = 1 \mid \mathrm{v}; \theta\right) = \sigma\left(c_j + \sum w_{ij} v_i\right) \tag{1}$$

where $\theta = \{w_{ij}, c_j, b_i\}$ and represents the model parameters, σ(x) is the sigmoid function, w_{ij} is the weight matrix between the visible layer and the hidden layer and c_j is the bias of the hidden layer.

Similarly, once the hidden units are computed, the zero-mean Gaussian activation of the visible layer can be recomputed as follows:

$$P\left(v_i = 1 \mid \mathrm{h}; \theta\right) \leftarrow \mathcal{N}\left(b_i + \sum_j w_{ij} h_i, \sigma^2\right) \tag{2}$$

where, b_i is the bias of the visible layer and $\mathcal{N}(\cdot)$ denotes a Gaussian distribution with zero-mean μ and variance σ^2.

As proposed in [8], the contrastive divergence (CD) algorithm can be used to build the model. In other words, utilizing an unsupervised learning strategy, the model parameters are learned by training the

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

first RBM layer based on a CD learning technique and using the normalized tiny images as stated before. As illustrated in Figure 3 (left), CD learning starts by setting the states of the visible units to a training vector. Then, the binary states of the hidden layer are computed in parallel using Equation 1. Once binary states have been sampled for the hidden units, a "reconstruction" is produced by setting each **v** to 1 with a probability given by Equation 2. Therefore, the model parameters can be updated using the following equations:

$$w_{inc\_ij} = \mu * w_{inc_{ij}} + \eta * \left(\left(\left((v_0 * h_0) - (v_1 * h_1)\right)/\gamma\right) - \lambda * w_{ij}\right.$$

$$w_{ij} = w_{ij} + w_{inc_{ij}} \quad (3)$$

$$b_i = b_i + \eta * (v_0 - v_1) \quad (4)$$

$$c_j = c_j + \eta * (h_0 - h_1) \quad (5)$$

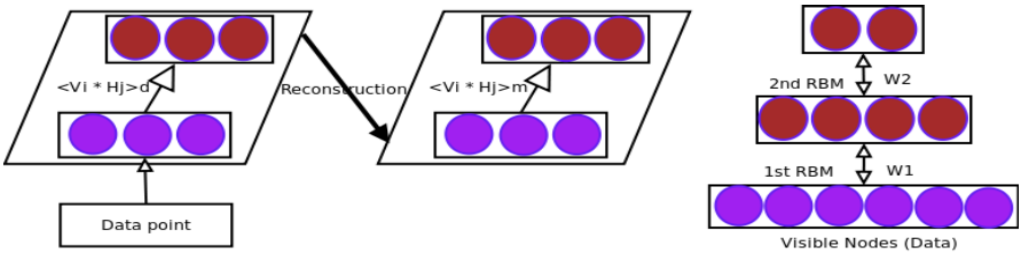


Figure 3. **Left**: Layer-wise training for an RBM with visible and hidden layers using contrastive divergence learning algorithm. **Right**: A deep belief network with two RBM layers.

The convergence of the network is achieved once the difference between the data statistics and the statistics of its representation generated by Gibbs sampling approaches zero, where the training dataset is fed to the network over the epochs. After the convergence of the network, a simple linear classifier, like softmax, is finally used to perform the classification process in the feature space and new samples from the validation dataset are used. It has been shown in many recent studies [8]-[9], [38] that using a deep learning approach plays important roles in: (1) Reducing the dimensionality of the data by using the most significant features to represent an object thus speeding up further tasks; the classification process for instance. (2) Improving the linear separability of the 28 signs of Arabic letters thus simplifying the overall classification process among them.

Therefore, the use of softmax regression was based on the assumption that the data becomes linearly separated in the feature space operated by DBNs. To underline this hypothesis, a non-linear classification algorithm, like SVM, will be used in the classification phase.

4. Experimental Results

4.1 Feature Extraction

Preliminary experiments have shown that the best structure for DBN training in terms of the final classification rate is the complete one (1024-1024). The training protocol is similar to the one proposed in [35] (300 epochs, a mini-batch size of 200, a learning rate of 0.02, an initial momentum of 0.5, a final momentum of 0.9, a weight decay of 0.0002, a sparsity target of 0.02 and a sparsity cost of 0.02).

After the network of the first RBM layer is converged, a set of sparse localized features were learned and extracted, as shown in Figure 4, by training the first RBM layer. One can see that these features represent most of the gestures of Arabic language letters. Some of the extracted features are very localized and represent small parts of the initial hands, like finger edges and hand borders and shapes. Another observation that can be mentioned here is that the thumb and/or index fingers can be seen in most of the extracted features which can be used later as reference features to code the testing images and perform the classification process. We have also seen that training a second RBM layer leads to reduce the classification rate, which might have suppressed some important features. Therefore, we assume that there is no need to train a second RBM layer, as the extracted features from the first RBM layer represent the shapes of most of the 28 signs of Arabic letters [35]. It can also be seen that some of the extracted features are overrepresented; for example, the signs of the following letters "NOON", "GHAYN" and "LAM". Some other features are underrepresented, such as the signs of the following letters "HA", "JIEM", "QAF", "FA" and "SAD". This might be due to the fact that the overrepresented signs have very sharp shapes which forced the learning network to extract them multiple times over other less sharp signs. The over-representation of high frequencies in the obtained feature will definitely play an important role in improving the linear separation of the initial data in the feature space thus enhancing the classification rate.

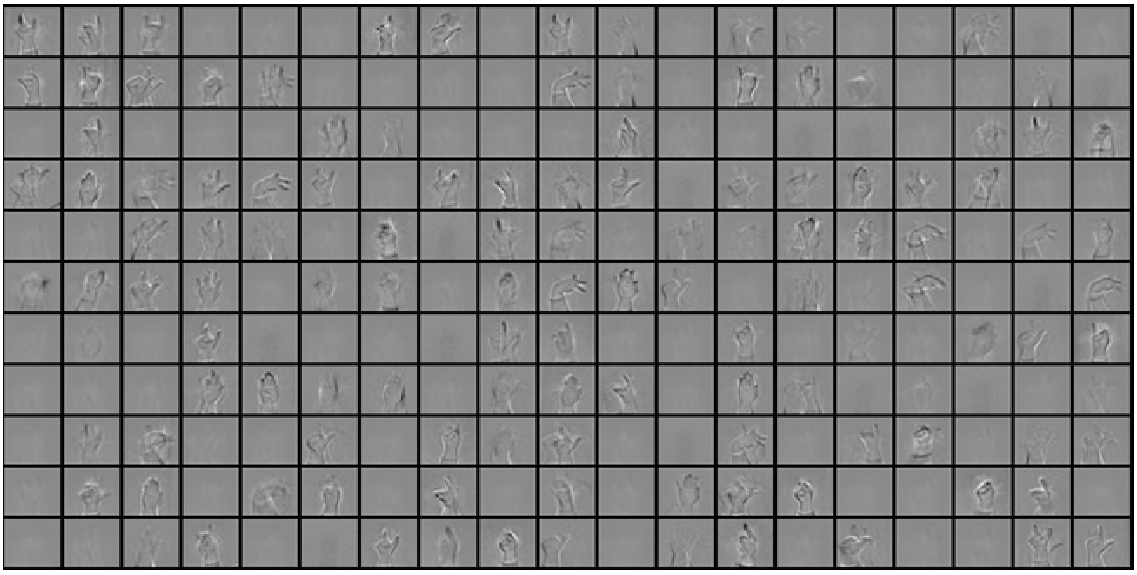


Figure 4. A sample of the 32×32 extracted features obtained by training the 1st RBM layer using the normalized tiny images. A detailed description of the training protocol and its training parameters used in this experiment can be found in [35].

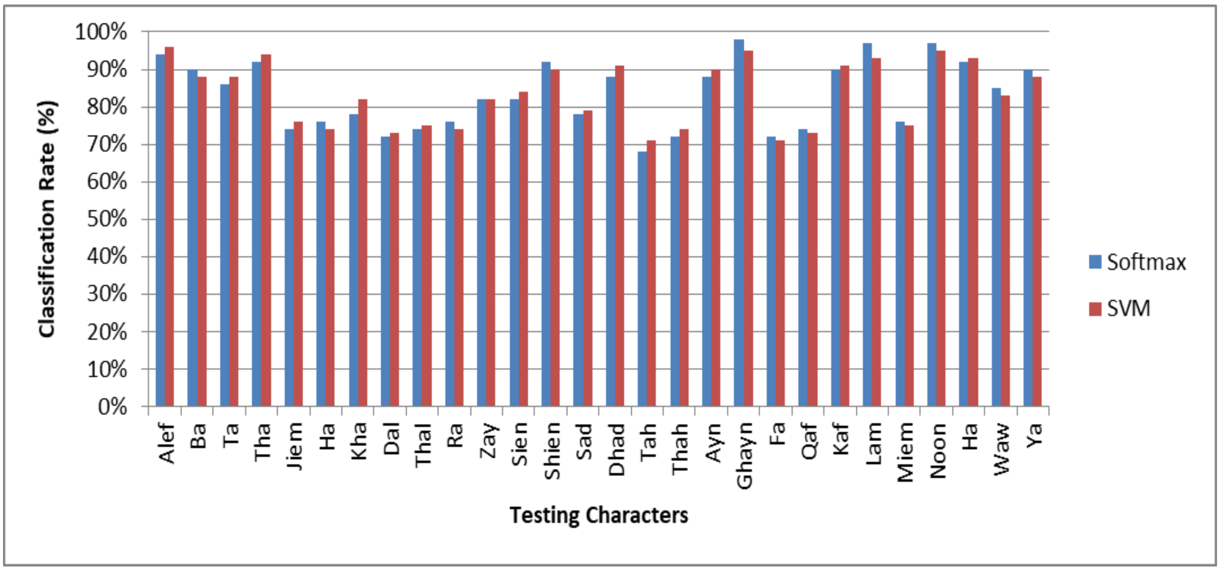


Figure 5. Average classification results for each letter sign of the Arabic language using DBNs coupled with tiny images and followed by softmax regression/ SVM.

4.2 Arabic Gesture Recognition

As mentioned before, after the learning phase and building an appropriate model using the extracted features, the next phase is the classification process for the testing dataset, which was created from the original dataset. The best classification results were achieved using a network of a single RBM layer,

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

as stated before. Therefore, the real-valued output of the first RBM units is used as an input to a softmax regression to perform the classification process. Figure 5 shows the classification results obtained using DBNs coupled with tiny images and followed by a softmax regression method. The use of a simple classifier was based on the assumption that after an appropriate image coding process is performed using the extracted features shown in Figure 4, the coded images become linearly separated.

For each image from the testing dataset, the softmax network uses the coded sign units to compute the probability of being one of the 28 Arabic letters. Based on the maximum probability value, the system identifies the corresponding character. As illustrated in Figure 5, the correct classification rate for each character of the Arabic language was ranging from 70% to 98% using the proposed model. The overall average of correct classification results for the 28 characters was 83.32% and 83.5% using a softmax regression and an SVM, respectively. These results are consistent and quite comparable when a sophisticated classification algorithm, like SVM, was used instead of softmax. This underlines and demonstrates that the use of DBN has significantly contributed to improving the linear separation between the gestures of Arabic language characters and thus a simple linear classifier was sufficient in the classification stage. These results are also quite comparable to the best recently published image-based approach [27], when the dataset is equally divided for training and testing the model.

5. DISCUSSION AND CONCLUSIONS

In this paper, a simple alternative approach based on deep learning and a linear classifier was proposed to classify the signs of Arabic letters and achieve accurate results. The overall obtained results of this proposed model are comparable to some of the existing approaches; for instance, see [23], [27] based on more complicated learning techniques and sophisticated classifiers and outperformed the results obtained in [29] based on various visual descriptors followed by an SVM. However, these results are still relatively lower than those obtained based on the use of hand-engineered signatures, like SIFT descriptors, followed by the use of a sophisticated classifier, like SVM [38]. Based on the obtained classification results shown in Figure 5, several observations can be discussed. It can be seen that the correct classification rate for some gestures; for example, the letters "NOON", "GHAYN" and "LAM", is high and reached 98%, while the accuracy for other gestures; for example, the letters "HA", "JIEM", "QAF", "FA" and "SAD", was ranging from 70% to 75%. This confirms that the representation of gestures in feature extraction plays an important role in achieving accurate results. It can also be seen that the correct classification rate for the following pair of gestures {("DAL", "THAL"), ("TAH", "THAH") and ("RA", "ZAY")} was ranging from 70% to 80%. The misclassification rate is attributed to the fact that each pair of these letters has strong similarities of gestures, which has probably forced the learning network to extract similar features thus complicating their classification process.

The final result is also illustrated in Table 1, where the recognition evaluation system parameters have been calculated for both models; DBNs followed by softmax regression and DBNs followed by an SVM. The results of these parameters are somehow similar and demonstrate that the linear separation of the initial data has been gained by DBNs and the use of the classification algorithm later will not significantly affect the result. In other words, the use of a simple classifier, like softmax regression, was therefore sufficient to obtain comparable classification results and the use of a sophisticated classifier, such as SVM, slightly improved the classification rate from %83.2 to %83.5. Also, the mean specificity of the proposed model was less when we used softmax regression compared to SVM, which indicates that the true negative rate of the classification results is lower. In addition to that, the mean sensitivity of the proposed model was higher when we used the softmax regression instead of SVM, showing that the true positive rate of the classification results is higher. Finally, the use of

Table 1. Recognition system evaluation parameters for DBNs followed by softmax/ SVM.

Performance parameters/ Classification method	Accuracy	Error	Sensitivity	Specificity	Precision	F1 score
DBNs followed by a softmax	0.832	0.177	0.705	0.962	0.955	0.811
DBNs followed by a SVM	0.835	0.175	0.675	0.995	0.993	0.804

DBNs followed by softmax regression achieved 81.1% F1 score, while f the use of DBNs followed by an SVM achieved 80.4% F1 score, which underlines that the proposed model is more simple and accurate. The major advantages of the proposed model can be summarized as follows. First, the obtained results demonstrated that a small image-based model followed by an appropriate feature space projection is capable of achieving comparable results to the recent methods [27], based on a complex learning technique (the use of convolutional neural network) and [38], based on a more sophisticated algorithm (the use of SIFT descriptors followed by the use of SVM classifier). Second, based on the obtained results, this paper presents a simple alternative to the existing approaches of ArSLR, by improving the linear separability of the initial data in the feature space thus simplifying the overall classification process. Third, after projecting the gestures onto an appropriate feature space that increases the linear separation between them the use of a nonlinear classifier, like SVM, did not change or improve the results.

Different ways will be investigated in the future to improve the results, including: (1) Assuming sharp signs to force the learning network to extract them multiple times. Thus, studying the effect of normalization and whitening on feature extraction remains an open question. (2) If we assume that extracting sparse features plays an important role in improving the classification results, then studying the sparsity factor with different parameters needs also to be considered. (3) Increasing the size of the dataset to ensure its scalability and to include new gestures that represent the Arabic digits for instance. (4) Increasing the number of images that were underrepresented in feature extraction and have similarities in gestures, thus studying their effect on the learning and classification phases.

REFERENCES

[1] M. Mohandes, M. Deriche and J. Liu, "Image-based and Sensor-based Approaches to Arabic Sign Language Recognition," IEEE Trans. Human-Machine Syst., vol. 44, no. 4, pp. 551–557, 2014.

[2] "Palestinian Central Bureau of Statistics," [Online], Available: http://www.pcbs.gov.ps/site/lang__ar/507/default.aspx#م.

[3] M. A. Abdel-Fattah, "Arabic Sign Language: A Perspective," Journal of Deaf Studies and Deaf Education, vol. 10, no. 2, pp. 212–221, 2005.

[4] A. Tiwari, A. B. Narayan and O. Dikshit, "Deep Larning Networks for Selection of Persistent Scatterer Pixels in Multi-temporal SAR Interferometric Processing," arXiv Prepr. arXiv …, 2019.

[5] R. Naoum, H. Owaied and S. Joudeh, "Development of A New Arabic Sign Language Recognition Using k-nearest Neighbor Algorithm," J. Emerg. Trends Comput. Inf. Sci., vol. 3, no. 8, pp. 1173–1178, 2012.

[6] N. Saliza et al., "Sign Language to Voice Recognition: Hand Detection Techniques for Vision-based Approach, " Current Developments in Technology-assisted Education, vol. 422, pp. 967-972, 2006.

[7] M. A. Mohandes, "Recognition of Two-Handed Arabic Signs Using the CyberGlove," Arab. J. Sci. Eng., vol. 38, no. 3, pp. 669–677, 2013.

[8] A. Hasasneh, E. Frenoux and P. Tarroux, "Semantic Place Recognition Based on Deep Belief Networks and Tiny Images," Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics (ICINCO-2012), vol. 2, pp. 236–241, 2012.

[9] A. Hasasneh, N. Kampel, P. Sripad and N. J. Shah, "Deep Learning Approach for Automatic Classification of Ocular and Cardiac Artifacts in MEG Data," vol. 2018, pp. 15–17, 2018.

[10] A. Hasasneh, Y.-A. Daraghmi and N. M. Hasasneh, "Towards Accurate Real-Time Traffic Sign Recognition Based on Unsupervised Deep Learning of Spatial Sparse Features: A Perspective," Int. J. Comput. Inf. Technol., vol. 06, no. 01, pp. 32–36, 2017.

[11] L. Liu et al., "Deep Learning for Generic Object Detection: A Survey," International Journal of Computer Vision, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[12] J. Cornejo and H. Pedrini, "Bimodal Emotion Recognition Based on Audio and Facial Parts Using Deep Convolutional Neural Networks," Proceedings of the 18th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 111–117, Dec. 2019.

[13] Y. C. Wong, L. J. Choi, R. S. S. Singh, H. Zhang and A. R. Syafeeza, "Deep Learning-based Racing Bib Number Detection and Recognition," Jordanian Journal of Computers and Information Technology

(JJCIT), vol. 5, no. 3, pp. 181–194, Dec. 2019.

[14] M. Elleuch, N. Tagougui and M. Kherallah, "Optimization of DBN Using Regularization Methods Applied for Recognizing Arabic Handwritten Scripts," Procedia Computer Science, vol. 108, pp. 2292–2297, Jan. 2017.

[15] C. Wan, "Research on Computer Information Retrieval Based on Deep Learning," IOP Conference Series: Materials Science and Engineering, vol. 677, no. 3, DOI: 10.1088/1757-899X/677/3/032101 2019.

[16] Z. Zhao, S. Wang, C. Jia, X. Zhang, S. Ma and J. Yang, "Light Field Image Compression Based on Deep Learning," Proceedings of IEEE International Conference on Multimedia and Expo (ICME), vol. 2018-July, San Diego, CA, USA, Oct. 2018.

[17] A. M. Riad, H. K. Elmonier, S. M. Shohieb and A. S. Asem, "SignsWorld: Deeping into the Silence World and Hearing Its Signs (State-of-the-Art), " Int. J. Comput. Sci. Inf. Technol., vol. 4, no. 1, pp. 189–208, Feb. 2012.

[18] R. Anderson, F. Wiryana et al., "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output," Procedia Computer Science, vol. 116, pp. 441–448, 2017.

[19] M. Mohandes, M. Deriche and J. Liu, "Image-based and Sensor-based Approaches to Arabic Sign Language Recognition," IEEE Trans. Human-Machine Syst., vol. 44, no. 4, pp. 551–557, 2014.

[20] P. K. Vijay, N. Suhas, C. S. Chandrashekhar and D. K. Dhananjay, "Recent Developments in Sign Language Recognition: A Review," Int. J. Adv. Comput. Eng. Commun. Technol., pp. 21–26, 2012.

[21] N. Tubaiz, T. Shanableh and K. Assaleh, "Glove-based Continuous Arabic Sign Language Recognition in User-Dependent Mode," IEEE Trans. Human-Machine Syst., vol. 45, no. 4, pp. 526–533, Aug. 2015.

[22] P. Kumar, H. Gauba and P. Roy, "A Multimodal Framework for Sensor Based Sign Language Recognition," Neurocomputing, vol. 259, pp. 21–38, 2017.

[23] H. M. Zawbaa, A. Ella Hassanien, K. Nakamatsu, N. El-Bendary and M. S. Daoud, "ArSLAT: Arabic Sign Language Alphabet Translator, " Int. J. Comput. Inf. Syst. Ind. Manag. Appl., vol. 3, pp. 498–506, 2011.

[24] M. Deriche, S. Aliyu and M. Mohandes, "An Intelligent Arabic Sign Language Recognition System Using a Pair of LMCs with GMM Based Classification," IEEE Sens. J., vol. 19, no. 18, pp. 1–12, 2019.

[25] A. A. I. Sidig, H. Luqman and S. A. Mahmoud, "Transform-based Arabic Sign Language Recognition," Procedia Computer Science, vol. 117, pp. 2–9, Jan. 2017.

[26] N. B. Ibrahim, M. M. Selim and H. H. Zayed, "An Automatic Arabic Sign Language Recognition System (ArSLRS)," J. King Saud Univ. - Comput. Inf. Sci., vol. 30, no. 4, pp. 470–477, Oct. 2018.

[27] S. Hayani, M. Benaddy, O. El Meslouhi and M. Kardouchi, "Arab Sign Language Recognition with Convolutional Neural Networks," Proceedings of International Conference of Computer Science and Renewable Energies (ICCSRE 2019), DOI: 10.1109/ICCSRE.2019.8807586, Agadir, Morocco, Jul. 2019.

[28] S. Shohieb and A. M. Riad, "Dynamic Hand Gesture Recognition: Real Time *vs.* Offline Recognition Using Kinect," Advances in Computer Science and Engineering, vol. 11, pp. 39–47, 2013.

[29] R. Alzohairi, R. Alghonaim, W. Alshehri, S. Aloqeely, M. Alzaidan and O. Bchir, "Image-based Arabic Sign Language Recognition System," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 3, pp. 185–194, 2018.

[30] A. Addin I. Sidig, H. Luqman and S. A. Mahmoud, "Arabic Sign Language Recognition Using Optical Flow-based Features and HMM," Proc. of International Conference of Reliable Information and Communication Technology (IRICT), pp. 297–305, 2018.

[31] M. Elbadawy, A. S. Elons, H. A. Shedeed and M. F. Tolba, "Arabic Sign Language Recognition with 3D Convolutional Neural Networks," Proc. of the 8th IEEE International Conference on Intelligent Computing and Information Systems (ICICIS 2017), vol. 2018-January, pp. 66–71, Cairo, Egypt, 2017.

[32] M. Mohandes, "Arabic Sign Language Recognition," Proc. of Int. Conf. Imaging Sci. Syst. Technol., vol. 1, pp. 753–759, Las Vegas, Nevada, USA, 2001.

[33] M. Maraqa, F. Al-Zboun, M. Dhyabat and R. A. Zitar, "Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks," J. Intell. Learn. Syst. Appl., vol. 04, no. 01, pp. 41–52, 2012.

[34] M. Maraqa and R. Abu-Zaiter, "Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks," Proc. of the 1st International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2008), pp. 478–481, Ostrava, Czech Republic, 2008.

[35] A. Hasasneh and S. Taqatqa, "Towards Arabic Alphabet and Number Sign Language Recognition," Global Journal of Computer Science and Technology, vol. 17, no. 2, 2017.

[36] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[37] A. Hasasneh, Robot Semantic Place Recognition Based on Deep Belief Networks and a Direct Use of Tiny Images, Ph.D. Diss., Paris SUD Univ., vol. 2, 2012.

[38] A. Tharwat et al., "SIFT-based Arabic Sign Language Recognition System," Proc. of Afro-European Conference for Industrial Advancement, Springer, Cham, vol. 334, pp. 359–370, 2015.

ملخص البحث:

إن أفضل طُرق التواصل بين فاقدي السّمع والأشخاص العاديين هي لغة الإشارة أو ما يُعرف بإيماءات الأيدي. وفي المجتمعات العربية، فإنّ الأشخاص فاقدي السّمع والمختصين هم فقط الذين باستطاعتهم التعامل مع لغة الإشارة بالعربية؛ الأمر الذي يجعل مجتمع فاقدي السّمع ضيقاً ويصعب تواصلهم مع الأشخاص العاديين. بالإضافة الى ذلك، فإنّ دراسة مشكلة تمييز لغة الإشارة بالعربية اكتسبت اهتماماً في الآونة الأخيرة؛ الأمر الذي يؤكدّ الحاجة الى استقصاء طرقٍ أخرى لحلّ مثل هذه المشكلة.

تقترح هذه الورقة طريقة جديدة لتمييز أحرف اللغة العربية ورموزها باستخدام لغة الإشارة، وذلك بناءً على خوارزمية تعلُّم عميق غير مراقبة، وشبكة ثقة عميقة (DBN) متصلةٍ باستخدامٍ مباشر لصور بالغة الصِّغر: من أجل استخدامها في تمييز أحرف اللغة العربية وتصنيفها. وقد أسهم استخدام التعلُّم العميق في استخلاص أهم السِّمات التي تمثل بصورة متفرقة، وكان له دور في تبسيط مهمة التمييز على سبيل الإجمال.

استُخدم ما يقرب من 6000 عينة من الإشارات المتعلقة بأحرف اللغة العربية الـ (28)، بعد إجراء عملياتٍ تتعلق بتكييف الحجم والتطبيع: من أجل استخلاص السِّمات. وقد جرى استقصاء عملية التصنيف باستخدام انحدار (سوفت ماكس)، وتم الحصول على دقة إجمالية بلغت 83.32%: الأمر الذي يؤكد موثوقية النموذج المستخدم في تمييز أحرف اللغة العربية. أما حساسية النموذج ودقته النوعية، بلغتا 70.5% و 96.2%، على الترتيب.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

N- WAY COMPACT ULTRA-WIDE BAND EQUAL AND UNEQUAL SPLIT TAPERED TRANSMISSION LINES WILKINSON POWER DIVIDER

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

(Received: 27-May-2020, Revised: 15-Jul.-2020, Accepted: 3-Aug.-2020)

ABSTRACT

In this article, compact N-way Ultra Wide Band (UWB) equal and unequal split Wilkinson Power Dividers (WPDs) using exponentially λ/4 Tapered Transmission Line Transformers (TTLTs) are designed. First, 2-way WPDs are designed, simulated and then cascaded to get 4-way (equal and unequal split) and 8-way (equal split) UWB WPDs. 2- and 4-way (equal and unequal split) WPDs are fabricated and tested. The simulated and measured results of all the designed dividers are good in terms of insertion, return losses and group delay through UWB frequency band. The analysis of these dividers is carried out using the commercial ANSYS High Frequency Structure Simulator (HFSS) software package which is based on the Finite Element Method (FEM). Moreover, A MATLAB built-in function "fmincon.m" is used to find the optimum values of the three resistors chosen for perfect isolation. To validate the results, the simulation results are compared with the measured ones.

KEYWORDS

Ultrawide band (UWB), Wilkinson power divider (WPD), Tapered transmission lines (TTLs), N–way WPD, HFSS.

1. INTRODUCTION

Ultra-wideband (UWB) wireless communication is a revolutionary wireless technology that provides excellent opportunities in the modern wireless communication system due to its special characteristics, such as low cost, high data rate, small physical size and less power consumption [1]. In 2002, the unlicensed use of UWB frequency band (3.1 GHz - 10.6 GHz) was authorized by Federal Communications Commission (FCC) with a restriction on transmit power level to -41.3 dBm/ MHz, avoiding the interference with the coexisting narrow band frequency technologies, such as Wireless Fidelity (WiFi) operating under different rules that share the same bandwidth within the UWB frequency range. Each radio channel in UWB has more than 500 MHz or 20% bandwidth, depending on its centre frequency [2]. The special characteristics of UWB technology make it more beneficial in many applications, such as in military, security, civilian commerce and medicine [3]-[4]. Power dividers are essential devices that enable the RF power to be divided or combined within an environment and they are widely used in many wireless communication applications, such as antenna diversity, radar applications, antenna feeders and frequency discriminators [5]. Wilkinson Power Divider (WPD) is the most commonly used divider and it was proposed to overcome the matching and isolation problems of T-junction power dividers. Since WPD generally provides narrow band, many efforts were done to make it suitable for the recent requirements in UWB wireless communication applications. In [6], a compact UWB WPD was proposed using only one section based on exponentially λ/4 TTLTs. One stepped-impedance open-circuited, overlapped butterfly radial and delta stub was added to each branch of WPD to increase the bandwidth (UWB) in [7]. Two-section Uniform Transmission Line Transformers (UTLTs) and TTLTs were used to design UWB WPD with an equal split ratio in [8]-[9]. UWB WPD with four sections of circular bending shape (TLTs) optimized based on micro-Genetic Algorithm (micro-GA) was proposed in [10]. Authors in [11] designed UWB WPD using binomial multi-section matching transformer. Taper equation in [6] was used to design 2- and 3-way unequal split UWB WPD in [12] and [13], respectively. UWB 3:1 WPD was proposed in [14] using two sections of Asymmetric Coupled Transmission Lines (ACTLs) and one section of two different length TTLs. UWB WPD with

1. S. Saleh, W. Ismail and I. Sorfina are with School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Penang, Malaysia. Emails: sahar_saleh@student.usm.my, eewidad@usm.my and intan.sorfina@usm.my
2. M. Haizal is with Wireless Communication Centre, Universiti Teknologi Malaysia, Johor, Malaysia. Email: haizal@utm.my
3. M. Bataineh and A. Alzoubi are with Hijjwai faculty for Engineering Technology, Yarmouk University, Irbid, Jordan. Emails: mohbat@yu.edu.jo and asem@yu.edu.jo

improved Fractional Bandwidth (FBW) was obtained in [15] by adding a circular stub and an L-type parasitic short line. Recently, Chebyshev type bandpass filtering UWB (1 GHz -5 GHz) equal split WPD using 5-section transformers with Short Circuited Stubs (SCSs) was designed in [16] based on synthesis theory, where the number and positions of stubs are controlled. UWB WPD can be also integrated with tuneable Band Pass Filters (BPFs) [17]-[18] to switch between UWB and other coexisting narrow bands in multiband communication system to reduce its size (instead of using multiple antennas).

In this paper, a compact UWB WPD is designed based on the technique used in [6]. N-way WPDs are mostly used as a feeding network to antenna arrays at different frequency bands [19]-[22]. In this work, 2-way, 4-way (equal and unequal split) and 8-way equal split UWB WPDs are designed using TTLTs in Sections 2, 3 and 4, respectively. Finally, Section 5 demonstrates the conclusion of this work. The simulation in this paper is carried out using ANSYS High Frequency Structure Simulator (HFSS) software package.

2. COMPACT 2-WAY TTLS UWB WPD

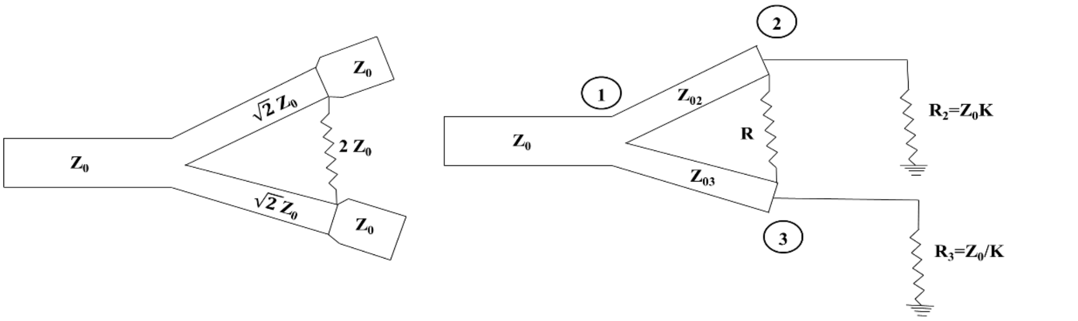
In the conventional equal split microstrip WPD, as shown in Figure 1a, the feeding port is connected to two parallel λ/4 Uniform Transmission Line Transformers (UTLTs) with a characteristic impedance of $\sqrt{2}\, Z_0$. The output ports are terminated with a microstrip transmission line having the same impedance as the line connecting the feeding port. The output ports are decoupled *via* a resistor R that equals 2 Z_0. Unequal split microstrip WPD is used in the design of a microwave distribution network to reduce the complexity of using a broadband coupler with a phase shifter. Unequal split power division is achieved if the impedances of λ/4 UTLTs are different from each other, as shown in Figure 1b. Furthermore, the second section of the quarter-wave transformers is needed to bring the arm impedance back to 50 Ω (this section is not included in Figure 1b for simplicity). In this figure, Z_{02} and Z_{03} are the characteristic impedances of the upper and lower arms, respectively. Here, $K^2 = P_3/P_2$ is the power splitting ratio between output ports 2 and 3 and $R = R'+R''$ is the isolation resistor between them. According to [5] ,

$$Z_{02} = K^2 Z_{03}\sqrt{K(1+K^2)} \tag{1.1}$$

$$Z_{03} = Z_0\sqrt{\frac{1+K^2}{K^3}} \tag{1.2}$$

$$R = R' + R'' = \frac{Z_0(1+K^2)}{K} \tag{1.3}$$

where $R' = KZ_0$ and $R'' = \frac{Z_0}{K}$



(a) (b)

Figure 1. WPD (a) Equal split (b) Unequal split.

Based on [6], each λ/4 UTLT in the equal-split and 2:1 unequal split WPD was replaced by its equivalent TTLT based on the optimum characteristic impedance profile of the transmission line:

$$ln\left(\frac{Z(z)}{Z_S}\right) = 0.5ln\left(\frac{Z_L}{Z_S}\right)\left\{1 + G\left[B, 2\left(\frac{z}{d} - 0.5\right)\right]\right\} \tag{1}$$

where Z_S and Z_L are the source and load impedances and d is the λ/4 TTL;

$$G(B,\xi) = \frac{B}{sinh(B)}\int_0^{\xi} I_0\left\{B\sqrt{1-\xi'^2}\right\} d\xi' \tag{2}$$

where $I_0(x)$ is the modified Bessel function of the first kind (zero order). B is a parameter chosen to minimize the internal return loss which is given by:

$$|R|_{max} = tanh\left[\frac{B}{sinh(B)}\ (0.21723)ln\left(\sqrt{\frac{Z_L}{Z_S}}\right)\right] \tag{3}$$

As B increases, lower input reflection is obtained; however, this will lead to a wide transmission line. So, B is selected to obtain a suitable return loss with a reasonable transmission line width. Here, B is selected to be 5.

Take in consideration that the operating frequency in this work is selected to be 3.1 GHz and the chosen substrate material is Rogers RO4003C with εr= 3.55, height h = 0.813 mm and dielectric loss tangent of 0.0027. Table 1 shows the dividers' design parameters**.** To find the optimum values of the three resistors in both equal and unequal split WPDs, the optimization process in [23] is applied. In this process, a MATLAB built-in function called "fmincon" is used, in which the transmission line of length d is divided into L sections according to the number of the required resistors. As the number of resistors increases, perfect isolation is achieved. In this work, three isolation resistors are chosen. Each tapered line is subdivided into M uniform short sections of length $\Delta z = d/M$. Then, each L section has M/L subdivisions. Odd analysis is used to find the required isolation resistors for equal-split UWB TTLWPD, as illustrated in Figure 2, where Z_{L2} and Z_{L3} are the load impedances of ports 2 and 3, respectively and Z_{in} is the input impedance looking into the output port**.**

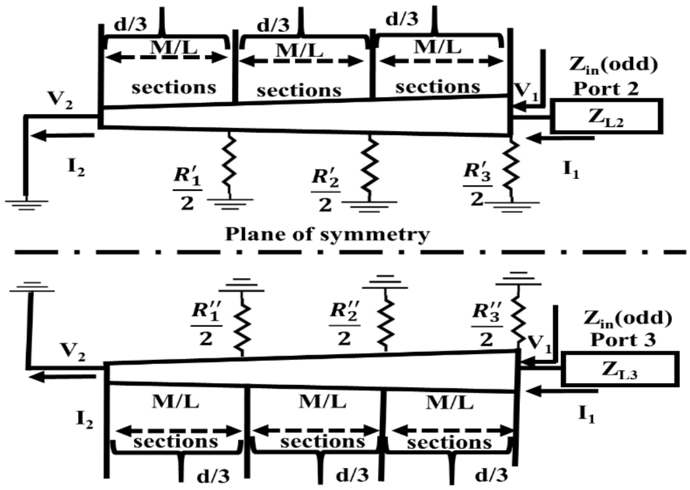


Figure 2. Odd-mode equivalent circuit for equal-split UWB TTLWPD.

To find the values of R_1, R_2 and R_3 and for perfect matching at the output, the following error function should be minimized *via* 'fmincon' function:

$$Error_{out} = \max\left(E_{f_1}^{out}, E_{f_2}^{out}, \ldots, E_{fm}^{out}\right) \tag{4}$$

where f_j (j = 1, 2, ….m) are the frequencies in UWB frequency band with Δf = 0.5 GHz and

$$E_{fj}^{out} = \left|\Gamma_{out}(f_j)\right|^2 \tag{4.a}$$

$$\Gamma_{out}(f_j) = \frac{Z_{in}^{o}\ (f_j) - Z_o}{Z_{in}^{o}\ (f_j) + Z_o} \tag{4.b}$$

And from Figure 2, $\frac{V_1}{I_1} = \frac{B}{D} = Z_{in}^{o}$; by setting $V_2 = 0$ and solving:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{Total} \cdot \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \tag{5}$$

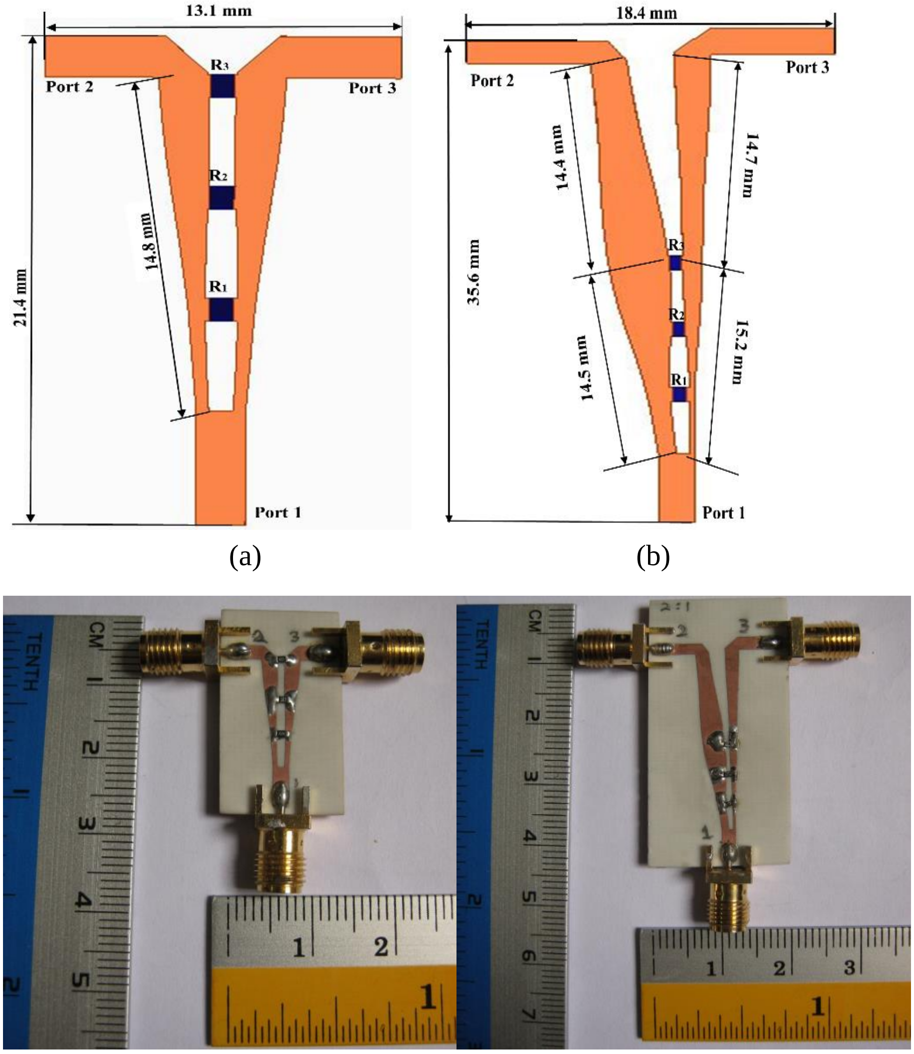
where,

$$[\text{ABCD}]_{Total} = [\text{ABCD}]_{\frac{R_3}{2}} \cdot [\text{ABCD}]_{1st\ section} \cdot [\text{ABCD}]_{\frac{R_2}{2}} \cdot [\text{ABCD}]_{2nd\ section} \cdot [\text{ABCD}]_{\frac{R_1}{2}} \cdot [\text{ABCD}]_{3rd\ section} \tag{5.a}$$

The same procedure is also applied to unequal-split UWB TTL WPD to find the required three resistors. Figure 3 shows the layouts and prototypes of the designed compact 2-way equal and 2:1 unequal split TTLs UWB WPDs.

Table 1. Calculated and optimized parameters for UWB equal and 2:1 unequal split WPD using tapered lines.

Parameters		**Z_S (Ω)**	**Z_L (Ω)**	**d (mm)**	**Resistors (Ω)**
Equal split		100	50	14.8	R_1=20, R_2= 120 and R_3=130
Unequal split	1st upper section	75	35.36	14.5	R_1=82 , R_2= 620 and R_3=470
	1st lower section	150	70.71	15.2	
	2nd upper section	35.36	50	14.4	
	2nd lower section	70.71	50	14.7	



(a) (b)

(c) (d)

Figure 3. (a) and (b) Configuration and (c) and (d) fabricated prototypes of the proposed compact 2-way equal and 2:1 unequal split TTLs UWB WPDs, respectively.

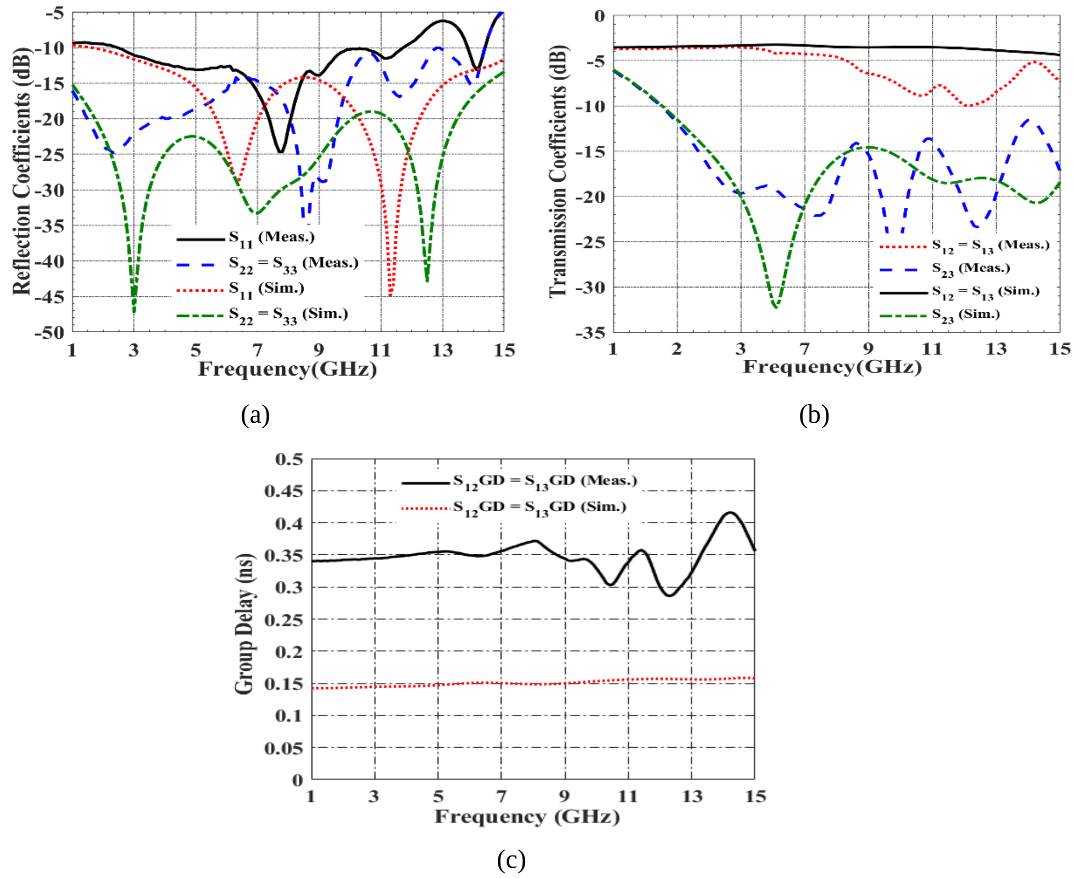


Figure 4. Measured and simulated (a) return loss (b) insertion loss and (c) group delay of the proposed 2-way equal split TTLs UWB WPD.

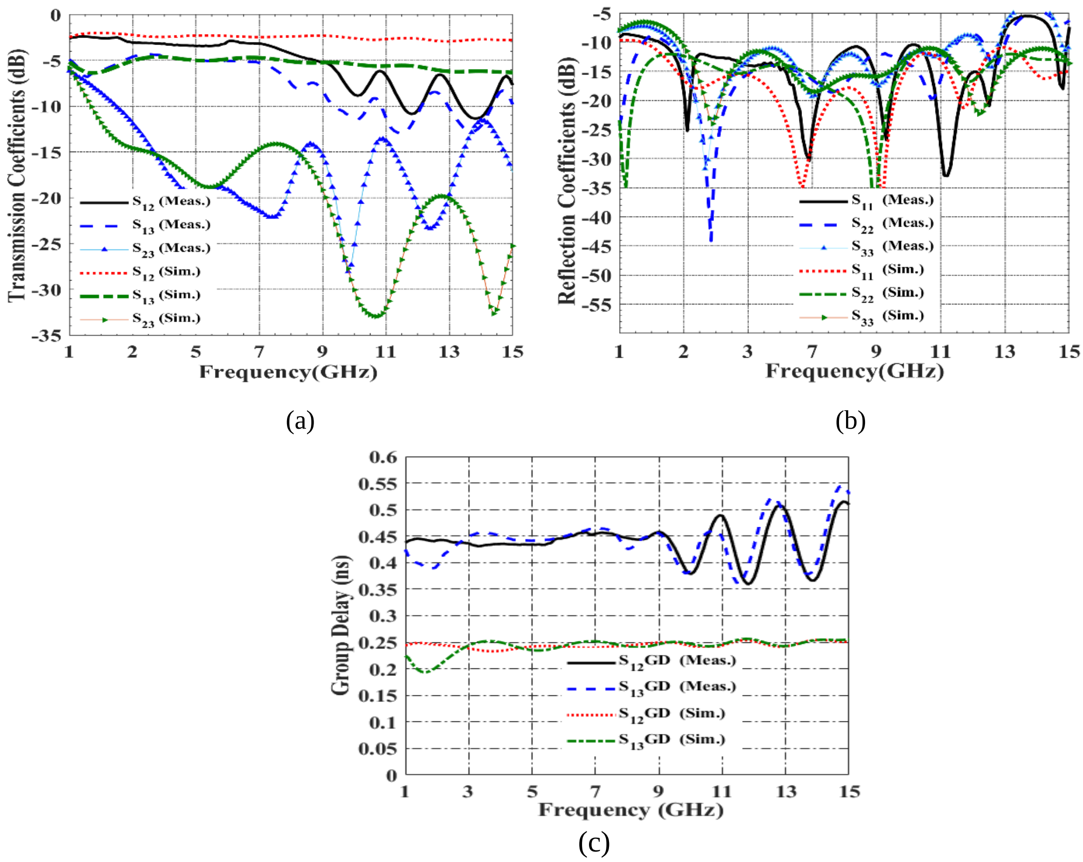


Figure 5. Measured and simulated (a) return loss (b) insertion loss and (c) group delay of the proposed 2-way 2:1 unequal split TTLs UWB WPD.

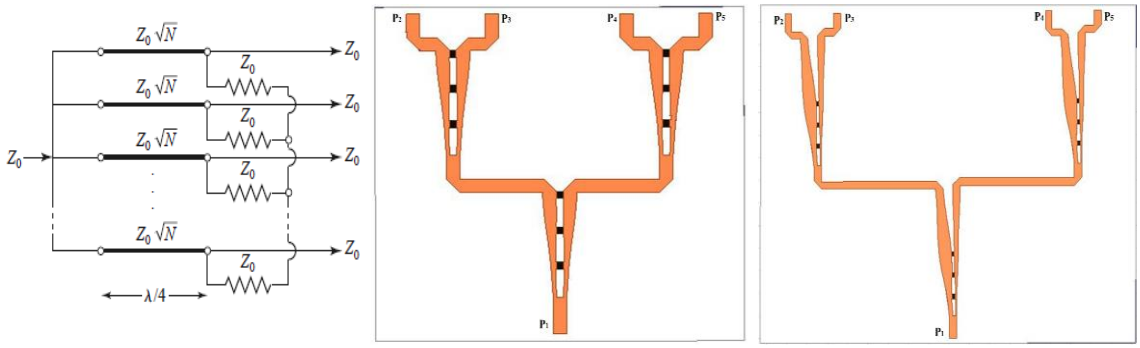
As it is clear from Figure 4a, the simulated and measured S_{11} is better than -10.7 and -10.1 through (1.6 GHz to more than 15 GHz) and (2.4 GHz -11.7 GHz), respectively. However, the simulated and measured $S_{22}= S_{33}$ is better than -18.7 and -10.4 through (less than 1 GHz to more than 15 GHz) and (less than 1 GHz -14.5 GHz), respectively. From Figure 4b, the simulated and measured transmission coefficients, $S_{12} = S_{13}$, are around -3 dB and -3 ± 0.8 dB, respectively with a disturbance in the measured one beyond 7.5 GHz because the impedance mismatch becomes worse at higher frequencies. In addition, for non-pure TEM transmission line at higher frequencies, the insertion loss increased due to the increase in conductor and dielectric losses [5]. Furthermore, a good isolation between the output ports is obtained (below -14 dB) for both simulated and measured results throughout the UWB frequency range. Figure 4c indicates constant group delays of approximately 0.15 ns (Sim.) and 0.35 ns (Meas.) and this difference is due to fabrication and measurement tolerance. However, for 2:1 unequal split WPD, as shown in Figure 5a, throughout the UWB frequency range, the simulated and measured S_{11}, S_{22} and S_{33} are below -10 dB. The simulated and measured S_{12} and S_{13} are around -2 dB and -5 dB and -2± 1.5 dB and -5± 0.9 dB, respectively with a degradation at high frequencies (beyond 7.5 GHz) for the measured results. Also, the simulated and measured isolation between output ports is better than -14 dB. Constant group delays of 0.26 ns (Sim.) and 0.45 ns (Meas.) are obtained, as indicated in Figure 5c. A comparison to the other UWB WPDs (equal and unequal split) in the literature is shown in Table 2.

3. COMPACT 4-WAY TTLS UWB WPD

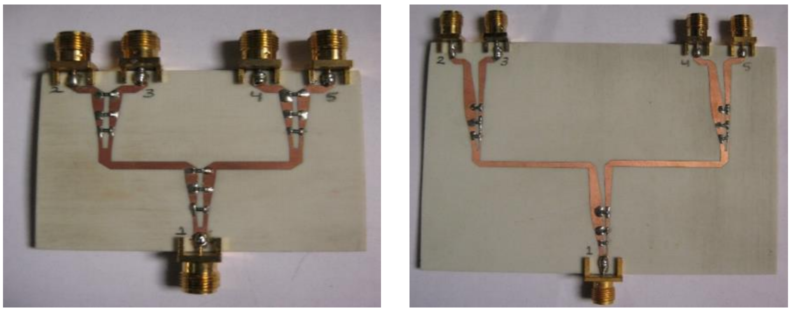
The output ports of WPD can be extended to N ports, as shown in Figure 6a. However, there will be a crossover for the resistors, which is difficult to realize, especially in planar technology. N-way WPD can also be obtained by cascading connection (stepped multiple sections), which avoids the resistors' crossover and this method is used in this paper to design 4- and 8-way WPDs. Although a design equation can be used to control the split ratio of multiple sections [24]-[26], for simplicity in this work, the same designed 2-way equal and 2:1 unequal split TTLs UWB WPDs in section 2 are used to design 4-way UWB WPD. In 4-way equal split UWB WPD, one fourth of the input power will be delivered to each one of the four output ports. However, for 4-way 2:1 unequal split UWB WPD, 2/3 of the input power will be delivered unequally (2:1) to ports 2 and 3; i.e., port 2 will have 4/9 of the input power and port 3 will have 2/9; in addition, 1/3 of the input power will be delivered to ports 4 and 5; i.e., port 4 will have 2/9 of the input power and port 5 will have 1/9. The layouts and prototypes of the proposed dividers are shown in Figure 6. For 4-way equal split WPD, Figure 7a shows good matching at the input and output ports; i.e., the simulated and measured S_{11} is better than -10.4 and -10.8 through (3.55 GHz – 13.3 GHz) and (3.55 GHz -12.2 GHz), respectively. In addition, $S_{22} = S_{33} = S_{44} = S_{55}$ (< -21 dB (Sim.) and < -13 dB (Meas.) throughout the UWB frequency range. From Figure 7b, the simulated and measured transmission coefficients, $S_{12} = S_{13} = S_{14} = S_{15}$ are equal to -6.02 ± 1 dB and -6.02 ± 1.6 dB, respectively with a degradation beyond 7GHz (for S_{15}) and 7.5 GHz, because at high frequencies, conductor and dielectric losses are increased for non-pure TEM transmission line. Figure 7c shows a constant simulated group delay of approximately 0.38 ns and due to the fabrication and measurement tolerance, the measured one is around 0.55 ns for all transmission coefficients. As noticed, this group delay is greater than that of 2-way equal split WPD because of the long path that the signal takes from port 1 to the output ports (2 stages). Furthermore, in Figure 7d, the simulated and measured isolation between the output ports $S_{23} = S_{45}$ and S_{34} is better than -13.8 dB and -20 dB, -14.4 dB and -16.3 dB, respectively throughout the UWB frequency range. However, for 4-way 2:1 unequal split WPD, the simulated and measured S_{11} is better than -11.1 dB and -11.2 dB through (3 GHz to more than 16 GHz) and (2.4 GHz -12GHz), respectively and S_{22}, S_{33}, S_{44} and S_{55} are < -10 dB, as depicted in Figure 8a and Figure 8b. The simulated and measured transmission coefficients in Figure 8c, S_{12}, $S_{13} = S_{14}$ and S_{15} are around -4 dB, -7 dB and -10 dB and -4 dB ± 1, -7 ± 0.7 dB and -10 ± 1dB, respectively, with a degradation beyond 7.5 GHz. The group delay in Figure 8d is around 0.6 ns (Sim) and 0.8 ns (Meas.), which implies the long path that the signal takes due to the other sections required for matching in both stages. Good isolation between the output ports is obtained, as shown in Figure 9e, throughout the UWB frequency range, where the simulated and measured $S_{23} = S_{45}$ and S_{34} are better than -10.3 dB and -20 dB and -22 dB and -25 dB, respectively.

Table 2. Comparison to related works in the literature.

Ref.	Substrate h(mm)/ ε_r	Technique used	Tr.(mm)	Rs	Fc (GHz)	BW (GHz)	Overall area (mm^2)	S_{11} (dB) <	S_{23} (dB) <	S_{12} (dB)	S_{13} (dB)
EQUAL SPLIT											
This work	**0.813/ 3.55**	**1 section of TTLs**	**14.8**	**3**	**3.1**	**2.4-11.7**	**≈ 21.4x13.1**	**-10.1**	**-14**	**≈- 3±0.8**	**≈-3±0.8**
[9] 2-way Single layer	0.508/ 2.2	2 sections TTLs and UTLs	TTL =4 & UTL =3	1	6.85	3.1-10.6	≈15.51x15.47	-11	-15	≈-3±0.6	≈-3±0.6
[10], 2-way Single layer	0.508/ 2.2	4 CUTLTs based on micro-GA	1st=4.646, 2nd=4.131, 3rd=3.386 & 4th =3.696	4	6.85	3.1-10.6	≈20.85x9.5	-14	-17	≈-3±0.5	≈-3±0.5
[8], 2-way Single layer	0.8/ 3.58	2 sections UTL and TTL	UTL =3.2 & TTL =11	2	7	3.1-10	≈21.1x6.3	-11	-12	≈-3±1.8	≈-3±1.8
UNEQUAL SPLIT											
This work	**0.813/ 3.55**	**2 sections of TTLTs**	**1st =15.2 2nd =14.7**	**3**	**3.1**	**2.3-12.9**	**≈35.6x18.4**	**-10.1**	**-14**	**≈-2±1.5**	**≈-5±0.9**
[12] 2-way 2:1, Single layer	0.1/ 2.33	2 sections of TTLTs	1st =27.5 2nd =27	5	2	2-12	76 x 28.5	-11.7	-15.5	≈-1.76±0.8	≈4.77±1
[13] 3-way 4:3:3, Single layer	0.635/ 2.33	2 sections of TTLTs	1st=27.5/28 2nd =27/28	4	2	2-12	≈59.18x 40	-10	S_{23} <-15 S_{24} & S_{34} <-10	≈-4	S_{13}=S_{14}≈-5.5
[14] 2-way 3:1, Single layer	0.508/ 2.33	ACTLs & TTLs	1st =15.9 2nd =35	2	NA	3.1-10.6	81.3. x 14	-12.3	-17	≈−1.42	≈−5.8

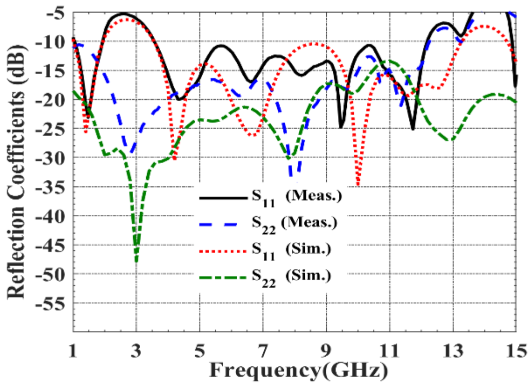


(a) (b) (c)

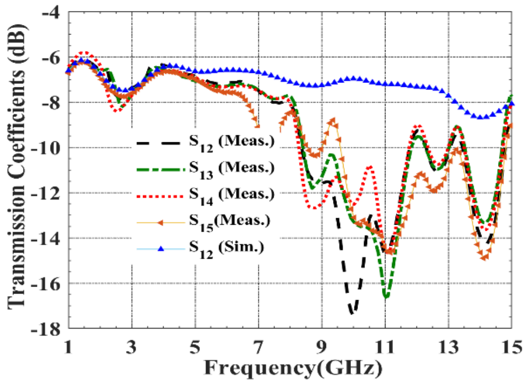


(d) (e)

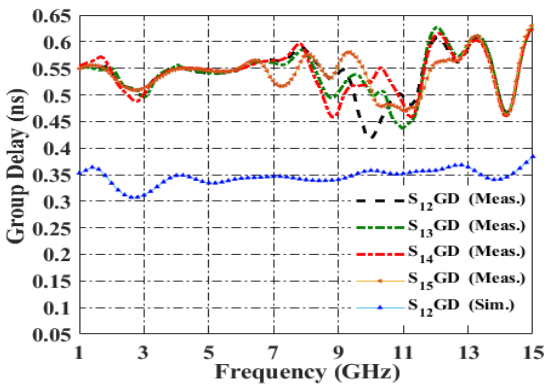
Figure 6. (a) N-way equal split WPD [5], (b),(c) Configuration and (d) fabricated prototypes of the proposed compact 4-way 2:1 equal and unequal split TTLs UWB WPDs, respectively.



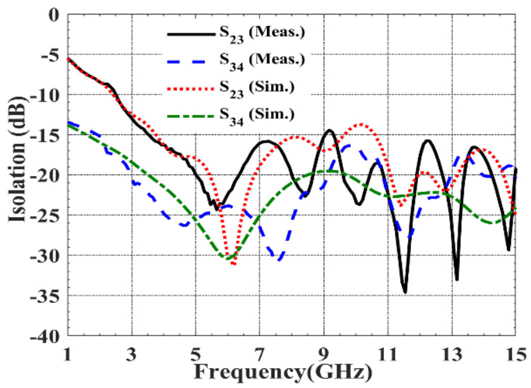
(a)



(b)



(c)



(d)

Figure 7. Measured and simulated (a) return loss (b) insertion loss (c) group delay and (d) isolation of the proposed 4-way TTLs UWB equal split WPD.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

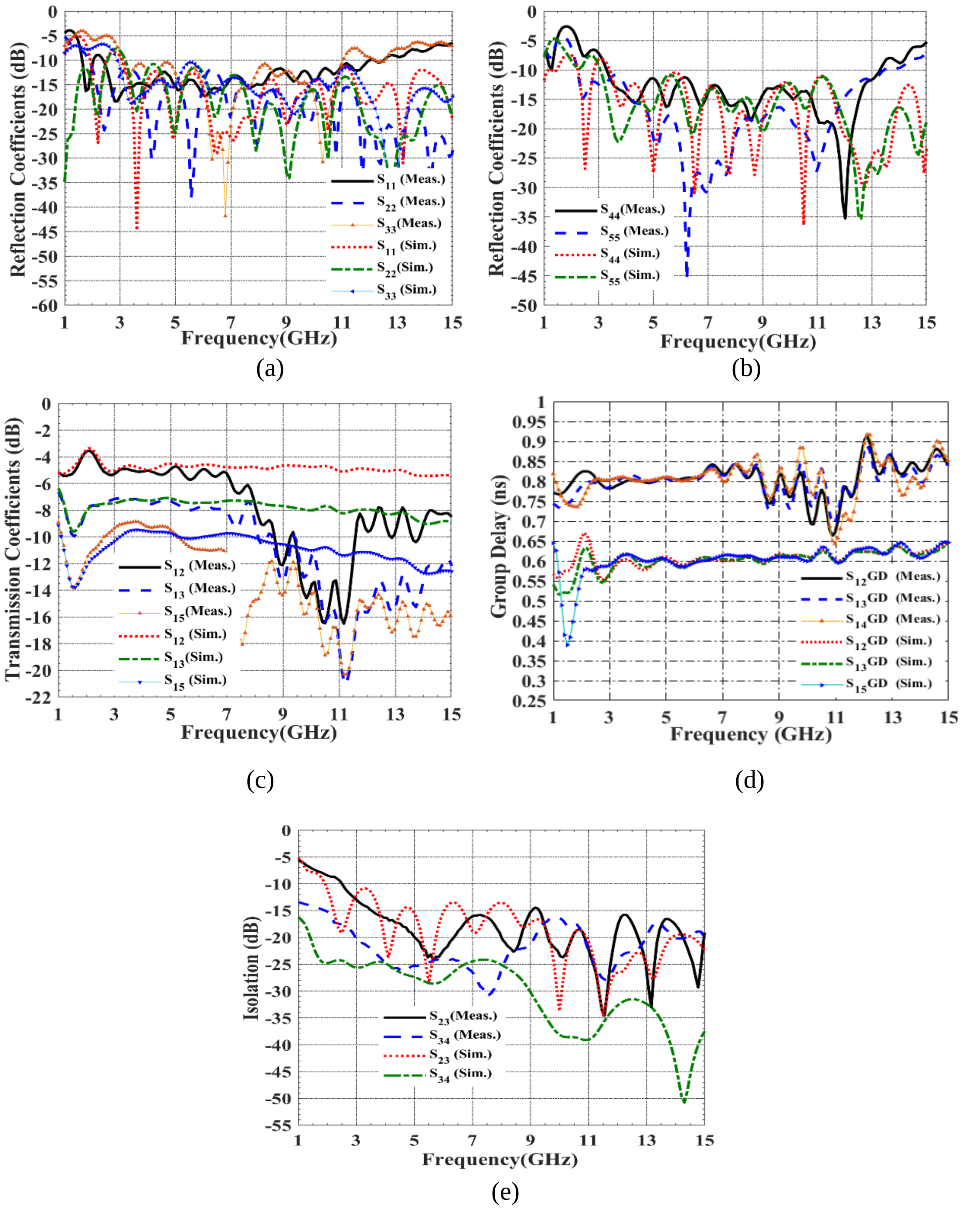


Figure 8. Measured and simulated (a), (b) return loss (c) insertion loss (d) group delay and (e) isolation of the proposed 4-way 2:1 unequal split TTLs UWB WPD.

4. COMPACT 8-WAY TTLs UWB EQUAL SPLIT WPD

An 8-way UWB equal split WPD is obtained by using three stepped sections with the help of the pervious designed 2-way equal split WPD. The first 2-way WPD is carrying two 2-way WPDs and each one is carrying two 2-way WPDs; at the end an 8-way WPD is obtained, as depicted in Figure 9. Each port will have 1/8 of the input power.

In Figure 10a, all output ports are matched in UWB frequency band and S_{11} is better than -10.2 dB through (3.9 GHz -13.4 GHz). The simulated transmission coefficients $S_{12} = S_{13} = S_{14} = S_{15} = S_{16} = S_{17} = S_{18} = S_{19}$ are equal to -9.03 ± 1.5 dB throughout the UWB frequency range. In Figure 10c, the group delay is around 0.6 ns for all the signal paths. Furthermore, the simulated isolation $S_{23} = S_{45} = S_{67} = S_{89}$ and $S_{34} = S_{56} = S_{78}$ is below -18.2 dB and -26 dB throughout the UWB band, as shown in Figure 10d.

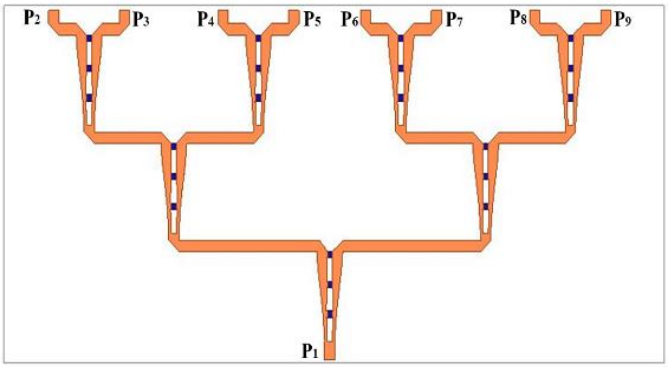
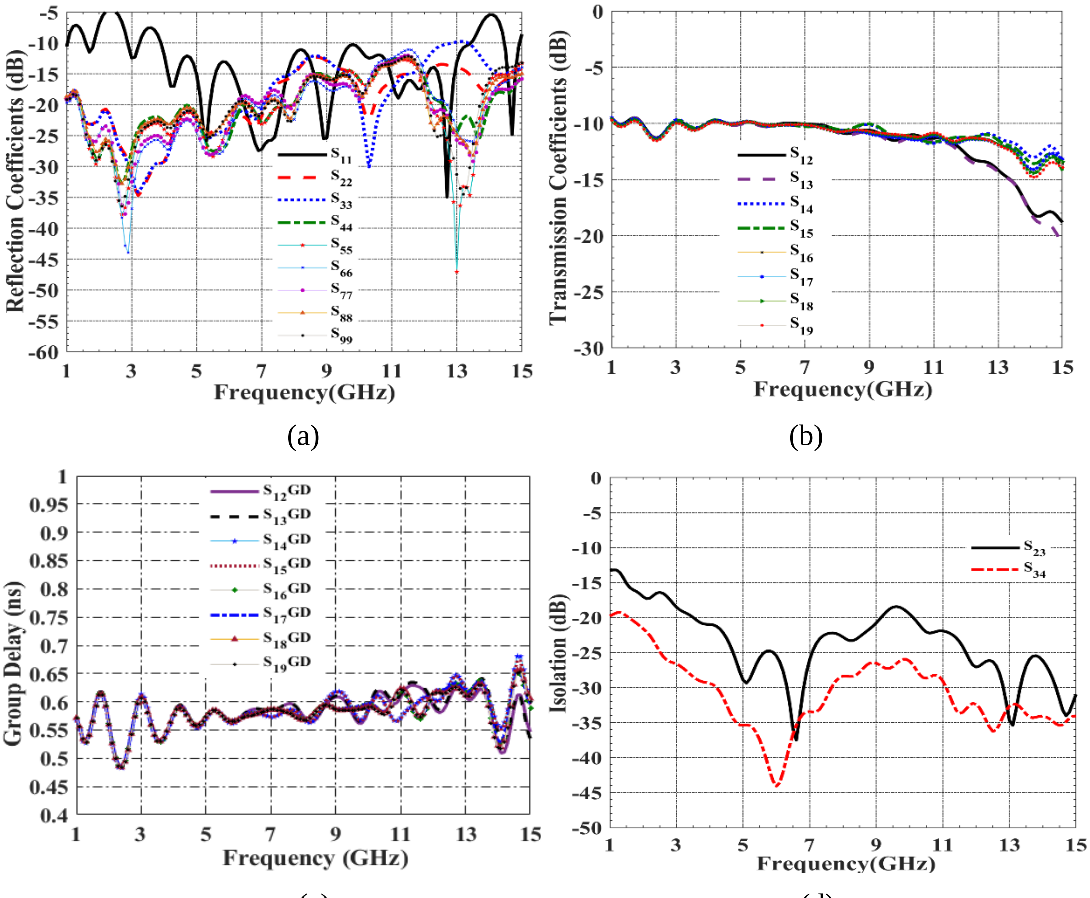


Figure 9: Layout of the proposed compact TTL UWB 8-way equal split WPD.



(a) (b)

(c) (d)

Figure 10. Simulated (a) return loss (b) insertion loss (c) group delay and (d) isolation of the proposed 8-way TTLs UWB equal split WPD.

5. CONCLUSIONS

Exponentially λ/4 Tapered Transmission Line Transformers (TTLTs) are used in this paper to design a compact N-way Ultra Wide Band (UWB) equal and unequal split Wilkinson Power Divider (WPD) with a cascaded topology. As a building block for N-way divider, 2-way equal and 2:1 unequal split TTLs UWB WPDs are designed, simulated and fabricated. Both dividers show good input and output matching, isolation and constant group delay. 2-stage 4-way equal split and 2:1 unequal split TTLs UWB WPDs are designed based on the designed 2-way dividers with good simulation and measured results in terms of reflection, transmission coefficients, isolation and group delay. Finally, based on 2-way equal split WPD, a 3-stage 8-way equal split WPD is designed with good simulation results. The proposed N-way WPD can be used as a feeding network for an antenna array. For future work, one can apply such different networks to linear UWB antenna arrays.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] V. Zhurbenko, Passive Microwave Components and Antennas, BoD–Books on Demand, 2010.

[2] R. S. Kshetrimayum, "An Introduction to UWB Communication Systems," IEEE Potentials, vol. 28, pp. 9-13, 2009.

[3] C. Nguyen and M. Miao, Design of CMOS RFIC Ultra-Wideband Impulse Transmitters and Receivers, ISBN 978-3-319-53107-6, Springer, 2017.

[4] Y. Rahayu, T. A. Rahman, R. Ngah and P. Hall, "Ultra Wideband Technology and Its Applications," Proc. of the 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN'08), pp. 1-5, 2008.

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

[6] C. T. Chiang and B.-K. Chung, "Ultra Wideband Power Divider Using Tapered Line," Progress In Electromagnetics Research, vol. 106, pp. 61-73, 2010.

[7] O. Ahmed and A.-R. Sebak, "A Modified Wilkinson Power Divider/Combiner for Ultra-wideband Communications," Proc. of IEEE Antennas and Propagation Society International Symposium, pp. 1-4, Charleston, SC, USA, 2009.

[8] H. Habibi and H. Miar Naimi, "Taper Transmission Line UWB Wilkinson Power Divider Analysis and Design," International Journal of Electronics, vol. 106, pp. 1332-1343, 2019.

[9] F. Razzak, M. Alkanhal and A.-F. Sheta, "UWB Wilkinson Power Divider Using Tapered Transmission Lines," Proceedings of Progress in Electromagnetics Research Symposium (PIERS), 2012.

[10] H. Wang, X. Tang, Y. Liu and Y. Cao, "Analysis and Design of Ultra-wideband Power Divider by Micro-genetic Algorithm," Journal of Electromagnetic Waves and Applications, vol. 26, pp. 1341-1349, 2012.

[11] B. Mishra, A. Rahman, S. Shaw, M. Mohd, S. Mondal and P. Sarkar, "Design of an Ultra-wideband Wilkinson Power Divider," Proc. of the 1st IEEE International Conference on Automation, Control, Energy and Systems (ACES), pp. 1-4, Hooghy, India, 2014.

[12] K. A. Al Shamaileh, N. I. Dib and A. Abbosh, "Analysis and Design of Ultra-wideband Unequal-split Wilkinson Power Divider Using Tapered Lines Transformers," Electromagnetics, vol. 32, pp. 426-437, 2012.

[13] D. Hawatmeh, K. Al Shamaileh, N. Dib and A. Sheta, "Design and Analysis of A 3-way Unequal Split Ultra-wideband Wilkinson Power Divider," Int. Journal of Electronics, vol. 100, pp. 1062-1071, 2013.

[14] Y. Wu and Y. Liu, "Compact 3–11 GHz UWB Planar Unequal Power Divider Using Two-section Asymmetric Coupled Transmission Lines and Non-uniform Microstrip," Electronics Letters, vol. 49, pp. 1002-1003, 2013.

[15] B. Pang and Y.-Z. Yin, "Development of Wilkinson Power Divider for UWB Application," Proc. of IEEE Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), pp. 1-3, Xuzhou, China, 2018.

[16] X. Wang, Z. Ma, T. Xie, M. Ohira, C.-P. Chen and G. Lu, "Synthesis Theory of Ultra-wideband Bandpass Transformer and Its Wilkinson Power Divider Application with Perfect in-band Reflection/Isolation," IEEE Transactions on Microwave Theory and Techniques, vol. 67, pp. 3377-3390, 2019.

[17] Y. S. Faouri, H. Sharif, L. Smadi and H. Jamleh, "Low Pass and Quad Band Pass Tunable Filter Based on Stub Resonator Technique," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 2, pp. 122-134, Aug. 2019.

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

[19] U. Ahmed and A. Abbosh, "Modified Wilkinson Power Divider Using Coupled Microstrip Lines and Shunt Open-ended Stubs," Electronics Letters, vol. 51, pp. 838-839, 2015.

[20] S. W. Wong and L. Zhu, "Ultra-wideband Power Divider with Good in-band Splitting and Isolation Performances," IEEE Microwave and Wireless Components Letters, vol. 18, pp. 518-520, 2008.

[21] Y. Yao and Z. Feng, "A Band-notched Ultra-wideband 1 to 4 Wilkinson Power Divider Using Symmetric Defected Ground Structure," Proc. of IEEE Antennas and Propagation Society International Symposium, pp. 1557-1560, Honolulu, HI, USA, 2007.

[22] Y. Yao, M. Liu, W. Chen and Z. Feng, "Analysis and Design of Wideband Widescan Planar Tapered Slot Antenna Array," IET Microwaves, Antennas & Propagation, vol. 4, pp. 1632-1638, 2010.

[23] S. Saleh, W. Ismail, I. S. Zainal Abidin, M. H. Jamaluddin, S. A. Al-Gailani, A. S. Alzoubi et al., "Nonuniform Compact Ultra-Wide Band Wilkinson Power Divider with Different Unequal Split Ratios," Journal of Electromagnetic Waves and Applications, vol. 34, no. 2, pp. 1-14, 2020.

[24] R. Mirzavand, M. M. Honari, A. Abdipour and G. Moradi, "Compact Microstrip Wilkinson Power Dividers with Harmonic Suppression and Arbitrary Power Division Ratios," IEEE Transactions on Microwave Theory and Techniques, vol. 61, pp. 61-68, 2012.

[25] M. M. Honari, L. Mirzavand, R. Mirzavand, A. Abdipour and P. Mousavi, "Theoretical Design of Broadband Multisection Wilkinson Power Dividers with Arbitrary Power Split Ratio," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 6, pp. 605-612, 2016.

[26] Y.-L. Wu, H. Zhou, Y.-X. Zhang and Y.-A. Liu, "An Unequal Wilkinson Power Divider for a Frequency and Its First Harmonic," IEEE Microwave and Wireless Components Letters, vol. 18, pp. 737-739, 2008.

ملخص البحث:

في هذه الورقة، يتم تصميم مقسِّمات قدرة من طراز ويلكنسون، مدمجة ومتعددة الطرق في نطاق الترددات فائق العرض، لاستخدامها في خطوط النقل المتدرجة المنفصلة بشكلٍ متساوٍ وغير متساوٍ والمبنية على أساس ربع طول الموجة. بدايةً، تم تصميم مقسِّمات قدرة ويلكنسون بطريقتين، ثم محاكاتها وسَلْسَلتها؛ للحصول على مقسِّمات قدرة ويلكنسون في نطاق الترددات فائق العرض ذات 4 طرق (بشكلٍ متساوٍ وغير متساوٍ) وذات 8 طرق (مقسمة بشكل متساوٍ). كذلك تم تصنيع مقسِّمات قدرة ويلكنسون ذات طريقين وذات 4 طرق (مقسَّمة بشكل متساوٍ وغير متساوٍ) وفحصها. وتبين أن نتائج القياس ونتائج المحاكاة لجميع المقسِّمات التي تم تصميمها كانت جيدة من حيث: الإدخال، وفقْد الرجوع، وتأخير المجموعة على مدى نطاق الترددات فائق العرض.

أُجري تحليل تلك المقسِّمات باستخدام حزمة برمجيات المحاكاة البنيوية للترددات العالية التابعة لبرمجيات (ANSYS) والمسماة (HFSS) التي ترتكز على طريقة العناصر المتناهية (FEM). من جهة أخرى، تم استخدام إحدى دوالّ (MATLAB) لإيجاد القيم المثالية للمقاومات الثلاث المختارة لتحقيق العزل التام. وللتحقق من النتائج، تمت مقارنة نتائج المحاكاة مع النتائج المقاسة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

PERFORMANCE ASSESSMENT OF THROUGHPUT IN A 5G SYSTEM

Mohammad A. T. Almahadeen and Aser M. Matarneh

(Received: 20-May-2020, Revised: 11-Jul.-2020 and 28-Jul.-2020, Accepted: 3-Aug.-2020)

ABSTRACT

This paper discusses the throughput of a fifth generation (5G) new radio (NR) system. The main goal of this research is to provide and develop a pathway for improving the throughput in the 5G system by investigating and controlling certain effective factors. The studied factors in this paper are the used modulation technique, the used subcarrier spacing in the default Clustered Delay Line (CDL) channel and the existence of a reflector in a custom CDL channel profile. It is found that the performance of the throughput is improved for larger subcarrier spacing and lower-order modulation technique. The existence and position of the reflector located between the transmitter and the receiver will be investigated relative to throughput performance in detail. Both fixed and changeable locations of the reflectors are considered in order to reach an optimal value of throughput. The results show that the existence of the reflector achieves a better throughput value compared to the one with no reflector. In the presence of a reflector and at a subcarrier spacing of 30 kHz, the throughput can reach 100% of throughput at 0 dB of signal to noise ratio (SNR) compared with only 40% at 0 dB for no-reflector case.

KEYWORDS

5G NR, Modulation, Reflector, Subcarrier spacing, Throughput.

1. INTRODUCTION

Challenges are the essential part of the technological development; thus, like all innovations, 5 G still has great challenges to address. As researchers have found, there is a very rapid growth in the advancement of radio technology. The travel is only around 40 years old (1G in 1980 and 5G in 2020) from 1G to 5G (Considering 5G in 2020). However, most of the information and communication technology (ICT)-based industries and consumers have recognized the value of improving the throughput, as the throughput is preferable to be as good as possible for many applications to ensure that the data is received appropriately [1]-[3].

5G has a potential and promises a compatible future, where it will give a much faster, wide range of applications and more reliable systems, without slowing down of running works. Self-driving cars, smart meters that track electricity usage and health-monitoring devices are just a few examples that may all take a big leap from what 5G could provide [4].

As a 5G trial experiment, an experiment was held in 2018 in Indonesia to test a 28 GHz system. This experiment aimed to test the characteristics of 5G mmWave band after implementing some scenarios in that 5G trial network [5]. One of these scenarios was made to determine the optimum coverage of that network and how it would be affected by changing some factors, such as the distance between the equipment of the network, obstacle existence and the reflector distance and angles to TUE (Test User Equipment) and AAU (Active Antenna Unit) [5]. Later, another experiment discussed the passive reflectors and how they can be functioned to enhance the coverage of the system for non-line of sight (NLOS) mmWave links.

Consequently, passive reflectors and some other metallic reflectors have shown significant results in terms of coverage enhancement for a new radio (NR) 28GHz system, as it used two shapes of reflectors, which are the spherical shape and the cylindrical shape and compared the results of both cases. Moreover, further calculations have been made regarding the gain of the reflectors, without ignoring one of the main factors in the reflection which is obviously the reflection angles, in addition to the distances between the reflector and both the transmitter and receiver [6].

M. A. T. Almahadeen and A. M. Matarneh are with Department of Electrical Engineering, Mutah University, Al-Karak, Jordan. Emails: mhammad2100@yahoo.com and aser.matarneh@mutah.edu.jo

The latest research held regarding the reflectors' effect discussed the possibility of employing reflectors to improve the coverage and to extend the range of the mmWave signal. That experiment introduced the use of ECHO passive reflectors and TURBO active repeaters to achieve an advancement of the coverage for both indoor and outdoor scenarios, taking into consideration the distance between the transmitter and receiver [7].

Accordingly, most of the past studies primarily focused on the whole system performance parameters without putting full focus on the throughput only, as this paper is concerned with. In this regard, the main goal of this research is to provide and develop a pathway to improve the throughput in a 5G system by investigating and controlling some effective factors, such as: modulation technique, subcarrier spacing (SCS) and existence of reflectors (position and location). Therefore, performance assessment will be carried out based on these factors for 3GPP communications standard.

According to the technical standard (TS) from the latest published version of the 3GPP TS 38.101 presented in 3GPP TS 38.101-1 V15.9.0 (2020-3) [8], the subcarrier spacing is not fixed in 5G system. SCS values which are used for the shared channels that carry traffic are 15 kHz, 30 kHz, 60 kHz and 120 kHz; whereas 240 kHz is used only for the synchronization signals. Also, the subcarrier spacing affects directly the number of the used resource blocks (RBs) for a given suggested bandwidth. Thus, it is interesting to investigate the significance of varying the SCS the throughput of the 5G system.

The second tested factor was the modulation technique which varied between the values (QPSK, 16QAM, 64QAM and 256QAM). These values affected the modulation order that is used in a formula which will be mentioned in the transport block size (TBS) in the methodology section.

The third and most important tested factor in this paper is the existence of a reflector. Passive metallic reflectors can be considered as a promising candidate for 5G systems due to many reasons coming from the fact that electromagnetic waves behave similarly to light.

The throughput is evaluated in case of a fixed distance between the transmitter and the receiver and in another case by moving the transmitter or the receiver; for example a driving user, which means that the distance will be changed between any two sides of the transmitter, receiver and reflector.

Table 1. List of abbreviations.

5G	Fifth Generation	HARQ	Hybrid Auto Repeat Request	QPSK	Quadrature Phase Shift Keying
AAU	Active Antenna Unit	ICT	Information and Communication Technology	RB	Resource Block
AoA	Angle of Arrival	IOT	Internet of Things	RTT	Round Trip Time
AoD	Angle of Departure	LOS	Line of Sight	Rx	Receiver
AWGN	Additive White Gaussian Noise	MIMO	Multi-input Multi-output	SCS	Subcarrier Spacing
CDL	Clustered Delay Line	NDI	New Data Indicator	SIB	System Information Block
CP	Cyclic Prefix	NLOS	Non Line of Sight	SNR	Signal to Noise Ratio
CRC	Cyclic Redundancy Check	NR	New Radio	SVD	Singular Value Decomposition
CSI	Channel Status Interference	NRB	Number of Resource Blocks	TBS	Transport Block Size
DL	Downlink	OFDM	Orthogonal Frequency Division Multiplexing	TUE	Test User Equipment
DL-SCH	Downlink Shared Channel	OOB	Out of Band	TX	Transmitter
DM-RS	Demodulation Reference Signal	PDSCH	Physical Downlink Shared Channel	UE	User Equipment
FR	Frequency Range	PRB	Physical Resource Blocks	QAM	Quadrature Amplitude Modulation

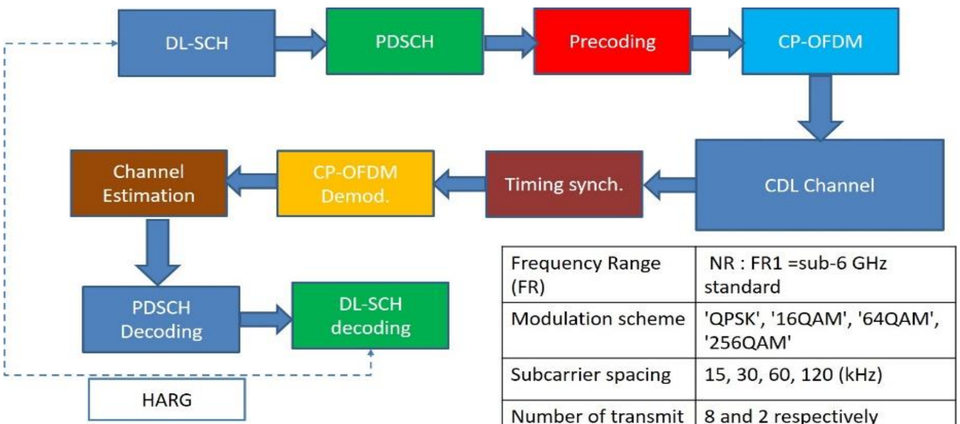
Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

This paper is organized as follows. Following this introduction, methodology is described in Section 2. Then, simulation results are presented and explained in detail in Section 3. Conclusions are then drawn in Section 4.

2. METHODOLOGY

The mechanism of evaluating the throughput of the communication standard 3GPP TS 38.101-1 5G-system went through the following procedure:

1. Setting up the parameters of the physical downlink shared channel PDSCH; this included the number of the resource blocks (RBs), Subcarrier spacing (SCS), the Signal to Noise Ratio (SNR) range, type of cyclic prefix (CP), number of allocated physical resource blocks (PRBs), number of PDSCH layers, the modulation technique, the code rate, the number of Hybrid Auto Repeat Request (HARQ) processes,…etc.
2. Updating the current HARQ process number, where the system will generate new data after checking the CRC of the previous transmission for that specific HARQ process and making sure that there is no need for retransmission.
3. Resource grid generation, in which the new radio downlink shared channel (nrDLSCH) will perform the channel coding, as its operation will be on the provided input transport block, while keeping a copy of the transport block to be retransmitted if needed; then, the precoding operation will be applied only on the resulting signal.
4. Waveform generation, where the generated grid will be OFDM-modulated.
5. Noisy channel modeling, where the waveform will be passed through a CDL fading channel and then the AWGN will be added; the SNR is defined per resource element (RE) for each user equipment (UE) antenna. In this step, for an SNR of 0dB, the signal and noise contribute equally to the energy per PDSCH RE per receive antenna [9].
6. Synchronization is done using the demodulation reference signal (DM-RS) and then, the synchronized signal is OFDM-demodulated.
7. Channel estimation is performed. In this experiment, perfect channel estimation is used. This assumption will be held in order to make the investigation of the tested factors more focused.
8. Calculating of the precoding matrix, which is done for the next transmission using singular value decomposition (SVD).
9. Decoding the PDSCH, where the recovered symbols are demodulated and descrambled by nrPDSCHDecode for all transmit and receive antenna pairs in order to obtain an estimate of the received codewords.
10. Decoding the downlink shared channel (DL-SCH) and storing the block CRC error for a HARQ process; the vector of decoded soft bits is passed to nrDLSCHDecoder which decodes the codeword and returns the CRC error to determine the throughput of the system.



Frequency Range (FR)	NR : FR1 =sub-6 GHz standard
Modulation scheme	'QPSK', '16QAM', '64QAM', '256QAM'
Subcarrier spacing	15, 30, 60, 120 (kHz)
Number of transmit and receive antennas	8 and 2 respectively (uniformly spaced)
SCs in each resource block	12

Figure 1. Flowchart of the simulation steps.

The flowchart in Figure 1 demonstrates the simulation steps mentioned above.

Evaluating the throughput depends basically on a value called the transport block size (TBS), which can be calculated using the guide in [10], as it has different formulae based on some factors affecting the parameters of N_{info} and coding rate, which both are the main factors of calculating the TBS.

$$N_{info} = N_{RE} \cdot R \cdot Q_m \cdot \upsilon \quad (1)$$

where, R is the code rate, Q_m is the modulation order: 2 for QPSK, 4 for 16QAM, 6 for 64QAM and 8 for 256QAM transmissions. υ is the number of layers [10]:

N_{RE} $=N_{RE} = min(156, N'_{RE}) \cdot n_{PRB}$; where, $N_{sc}^{RB} = 12$, which is the number of subcarriers in a physical resource.

$N'_{RE} = N_{sc}^{RB} \cdot N_{symb}^{sh} - N_{DMRS}^{PRB} - N_{oh}^{PRB}$; where, N_{symb}^{sh} : is the number of symbols of the PDSCH allocation within the slot, N_{DMRS}^{PRB}: is the number of REs for DM-RS per PRB in the scheduled duration including the overhead of the DM-RS CDM groups without data and N_{oh}^{PRB}: is the overhead configured by higher layer.

Evaluating the TBS can lead directly to the value of the maximum throughput, by multiplying the value of the TBS by the number of the slots in the system, which varies based on the numerology and the number of frames as in Table 4.3.2-1 in [11].

The procedure mentioned above is generally used for the throughput calculation, with changing the values of the factors that are intended to be investigated.

The third studied factor was the existence of a reflector with the custom delay profile CDL channel, as that mentioned in [12]. In this paper, the impact of the position of the reflector has been studied in case of fixed transmitter and receiver and in case of moving transmitter or receiver, with a reflector design as shown in Figure 2.

Spacing between reflector and both transmitter (Tx) and receiver (Rx) was calculated by assuming a side and angle of the triangle. After that, the remaining needed values were calculated to decide where to locate the reflector in the appropriate position to achieve the best possible performance as will appear in the results.

The reflector type preferred to be used will be the metallic one due to its advantages compared to the other types [13]. Furthermore, the reflection properties of electromagnetic waves are better at higher frequencies because of the smaller skin depth [14] and lower material penetration. Thus, metallic reflectors can act similarly as a communication repeater with the advantage that it can function without electricity in addition to negligible maintenance, with longer life spans and small initial investment costs compared to repeaters consisting of active elements.

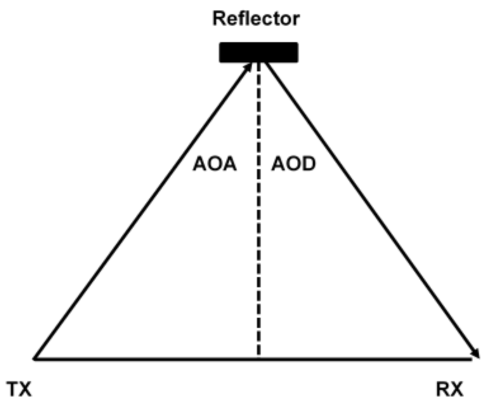


Figure 2. Reflector design.

These metallic reflectors can be part of our everyday objects, such as lamp posts, advertisement boards and street signs. For the characteristics of the metallic reflectors, power can be calculated using the following formula [6]:

$$P = P_{refl}^{(1)} + P_{refl}^{(1)} + P_{olos} + P_s \quad (2)$$

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 03, September 2020.

where $P^{(1)}_{refl}$ and $P^{(2)}_{refl}$ are the received powers due to first- and second-order reflections from the reflectors, respectively, P_{olos} is the power from the obstructed LOS (OLOS) path and P_s is the received power from other surrounding objects.

The transmitted power density at the reflector can be calculated using the following formula [6]:

$$P_{refl}(R_1) = \frac{P_{tx} G_{tx}(\theta_{tx}\ ,\ \emptyset_{tx})}{4\pi R_1^2} \tag{3}$$

The transmitted power density at the reflector is denoted as P_{refl} (R_1), at distance R_1 from transmitter, P_{tx} and $G_{tx}(\theta_{tx}, \varphi_{tx})$ are the transmitted isotropic power and gain (directivity) of the transmit antenna at respective azimuth and elevation angles of θ_{tx} and φ_{tx} [6].

3. Results and Discussion

In the following results, the simulations have been conducted using Matlab Platform.

3.1 Subcarrier Spacing

The subcarrier spacing values had varied between 15 kHz, 30 kHz, 60 kHz and 120 kHz, since the subcarrier of 240 kHz was not used, as it is usually used only for the synchronization signals and not for the shared channels which carry traffic. The results of changing the subcarriers were as illustrated in Figure 3.

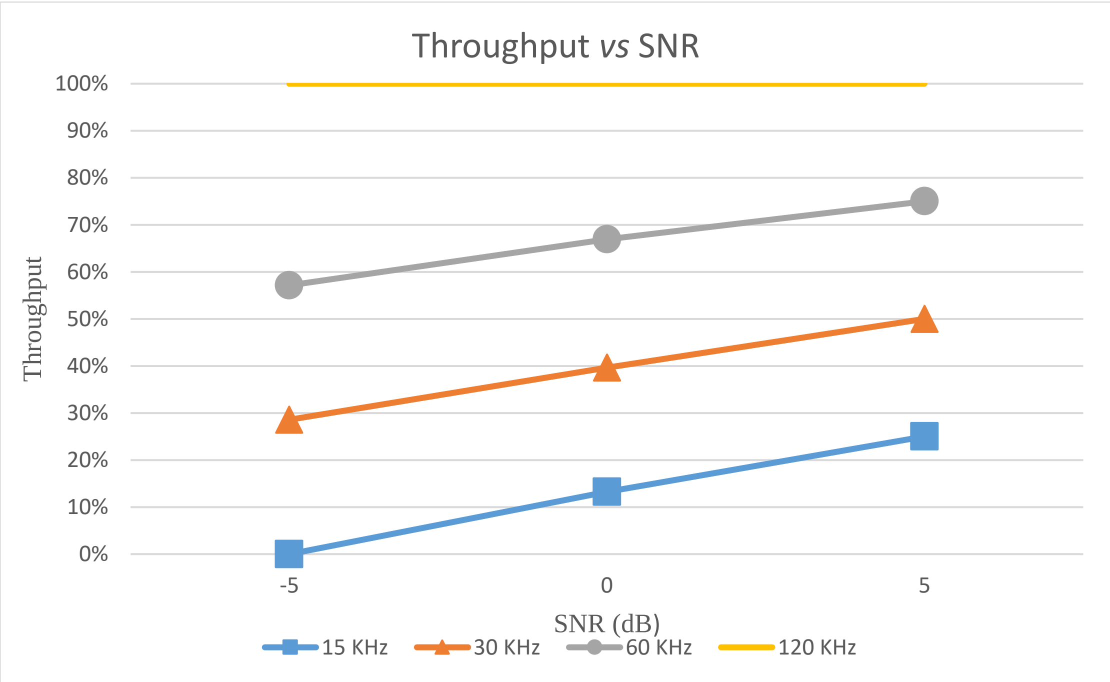


Figure 3. Throughput percentage at different SCS.

It can be noticed from Figure 3 that small subcarrier spacing allows more subcarriers to be available for a given amount of bandwidth, thus increasing the spectral efficiency, since more data is available for a given amount of bandwidth. However, performance degrades as subcarrier spacing decreases due to inter-channel interference [15].

Table 2. Throughput values for different SCS values.

SCS	Throughput (bps) at -5 dB	Throughput (bps) at 0dB	Throughput (bps) at 5dB
15 kHz	0	639520	3197600
30 kHz	639168	1278336	3195840
60 kHz	622976	1284888	3114880
120 kHz	910080	1516800	3033600

The throughput performance evaluation of the system has been varied by changing the applied subcarrier spacing of the system. Different to the previous generations of wireless mobile systems which had a fixed subcarrier spacing (mostly 15 kHz), 5G has the ability to change the subcarrier spacing value, where it can have the value of 15 kHz, 30 kHz, 60 kHz and 120 kHz. Accordingly, the used values of subcarrier spacing in the simulation are varied between 15 kHz and 120 kHz.

Starting from a subcarrier spacing of 15 kHz which produced a value of 12% of throughput at 0dB, the performance of the throughput improved as the values of the subcarrier spacing are increased, as shown by the values of 40%, 68.25% and 100% of throughput for subcarrier spacing's of 30 kHz, 60 kHz and 120 kHz, respectively, as shown in Figure 3. Moreover, for clarity purposes, the values of the throughput shown in Table 2 are calculated by multiplying the value of TBS by the number of slots in each case of SCS.

3.2 Modulation Techniques

For the modulation technique used, QPSK, 16QAM, 64QAM and 256QAM techniques had been tested at 30 kHz SCS and the results are shown in Figure 4. The throughput performance of the system was investigated with the modulation levels applied, where the modulation levels applied were: QPSK, 16QAM, 64QAM and 256QAM, with fixing the subcarrier spacing to 30 kHz.

A comparison between the performance of throughput with different types of modulation shows that the throughput performance can be improved further by using lower level of modulation. However, there is a trade-off between modulation techniques used due to their impact on the spectral efficiency [16]-[17], where using a lower-order modulation technique would improve the throughput performance but it would result in a lower spectral efficiency.

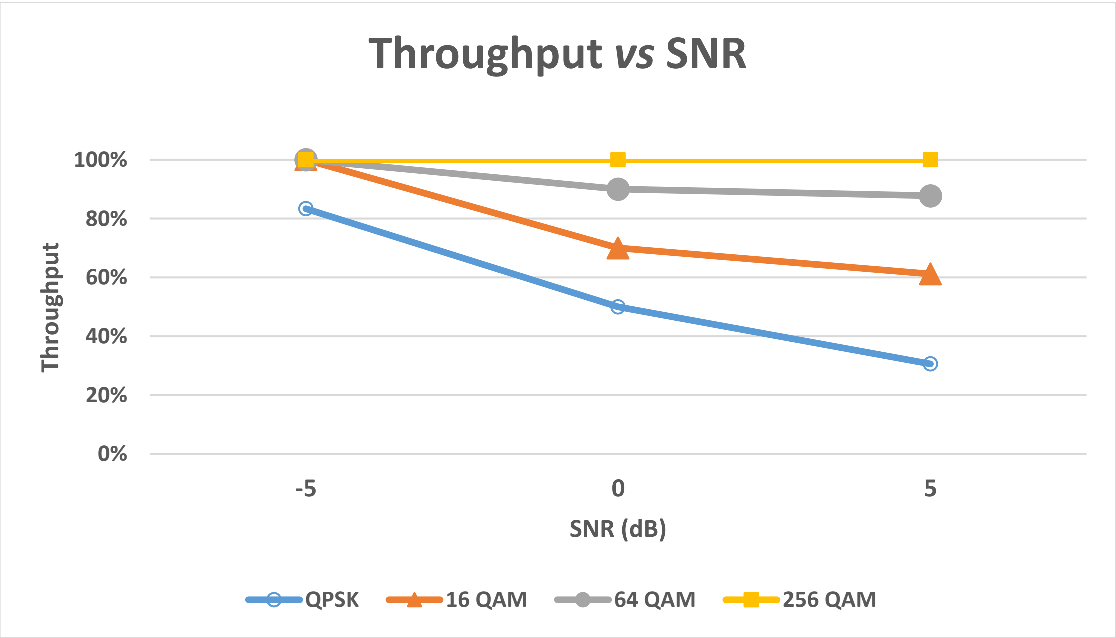


Figure 4. Throughput percentages of different modulation levels.

3.3 Existence of Reflector

Based on the previous results, a trade-off should be made to improve the throughput performance, either by decreasing the data rate and consuming more bandwidth in case of higher SCS, or decreasing the spectral efficiency to have a good throughput performance in case of lower modulation techniques.

Having a reflector could be a solution to have a balanced system. In this subsection, the existence of a reflector and how it can help improve the throughput performance will be discussed.

The default values of the CDL have been stated earlier in the previous results of throughput with changing either the subcarrier spacing or the modulation level, but for the custom values of CDL, the parameters which have been tested were the position and location of the reflector.

The existence of the reflector has shown significant results of throughput, where it could cause a much better performance or a much worse performance depending on the location and position of the reflector.

The existence of the reflector in the 5G system in this paper is investigated for two scenarios. First, a fixed distance between transmitter and receiver is assumed with changing the position of the reflector. Second, different distances between transmitter and receiver are investigated in the existence of a reflector, which might happen as one or more users are moving or driving for example.

In each scenario and for each position and location of the incorporated reflector, the angle of departure (AoD) and the angle of arrival (AoA) will be calculated and used in the simulation. Furthermore, the frequency range used in NR 5G system is FR1 which denotes a sub-6 GHz standard.

Therefore, the procedure for testing the reflector impact on the throughput was carried out through changing some parameters as follows:

1. Angle of Arrival (AoA)
2. Angle of Departure (AoD)
3. Distance from Transmitter to Receiver (Tx-Rx)
4. Distance from Transmitter to Reflector (Tx-Ref)
5. Distance from Receiver to Reflector (Rx-Ref), as will appear in each of the following tables.

In the case of fixed distance between Tx and Rx, this distance was assumed to be 16m and the distances between reflector, Tx and Rx were assumed for the first time and then adjusted closer and further till reaching the best possible scenario.

For the case of varying distance between Tx and Rx, the distances of the first scenario were assumed and then adjusted in different cases to have different scenarios with varied results of throughput performance. It should be noted that throughout this section, the SCS is fixed at 30 kHz and NRB=51.

a) Fixed Distance between Transmitter and Receiver

In this part, the separation between Tx and Rx was fixed to 16m and the position and location of the reflector were varied. The throughput results are shown in Table 3 for five cases and without reflector for comparison purposes.

Table 3. Different reflector positions for fixed 16m distance from Tx to Rx.

	Tx-Rx	**Tx-Ref.**	**Rx-Ref.**	**Throughput % at 0dB**	**Throughput % at 5dB**	**Difference % at 0dB**	**Difference % at 5dB**
No Reflector	-	-	-	42.5	100	-	-
Case 1	16	10	10	15	40	-27.5	-60
Case 2	16	15	15	40	100	-2.5	0
Case 3	16	12	20	97.5	100	55	0
Case 4	16	12	22	100	100	57.5	0
Case 5	16	18	8.5	0	0	-42.5	-100

i. Case 1: Tx-Rx = 16m, Tx-Reflector = Rx-Reflector = 10m

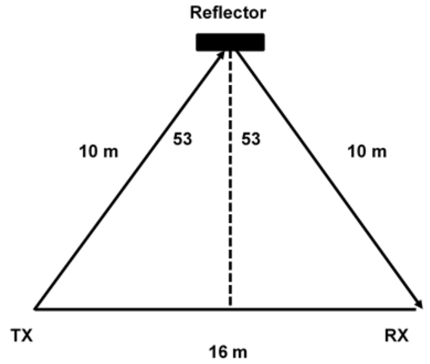
In this case, the throughput performance was below that in the case with no reflector, as the system achieved 15% of throughput at 0dB and 40% at 5dB. Figure 5 (a) and Figure 5 (b) show the reflector position and the throughput performance, respectively.

ii. Case 2, Tx-Rx =16m, Tx-Ref. = Rx-Ref. = 15m

In this case, the throughput performance was still below that in the case with no reflector, as the system achieved 40% of throughput at 0dB and 100% at 5dB, as depicted in Figure 6 (a) and Figure 6 (b), respectively.

iii. Case 3, Tx-Rx = 16m, Tx-Ref. = 12m, Rx-Ref. = 20m

In this case, the throughput performance was much better than in the case with no reflector, as the system achieved 97.5% of throughput at 0dB and 100% at 5dB, as depicted in Figure 7 which shows the reflector position and the throughput performance.



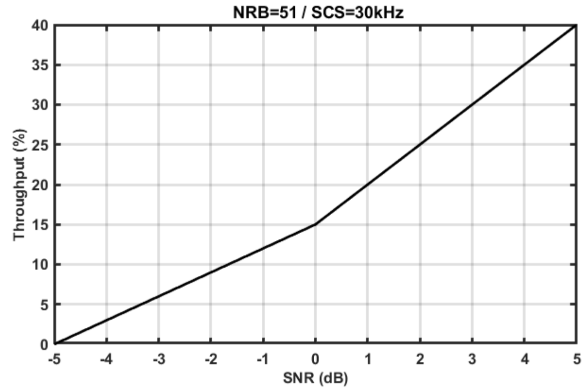
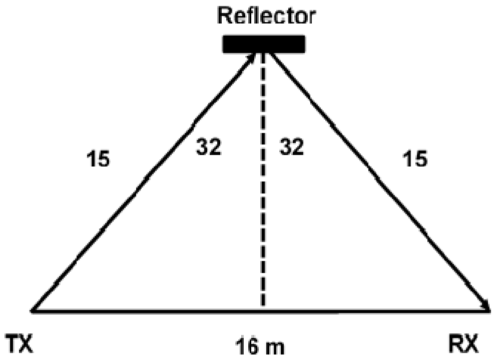


Figure 5 (a). Reflector design of case 1. (b) Throughput percentage for case 1.



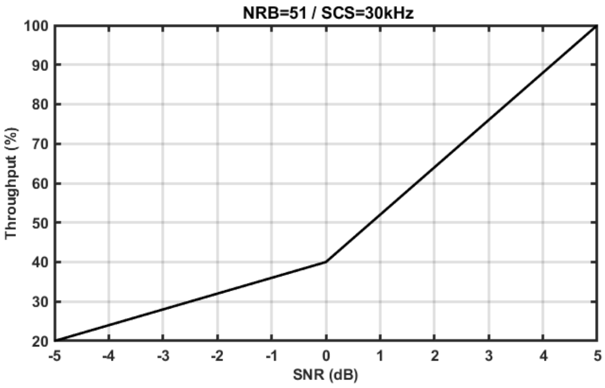
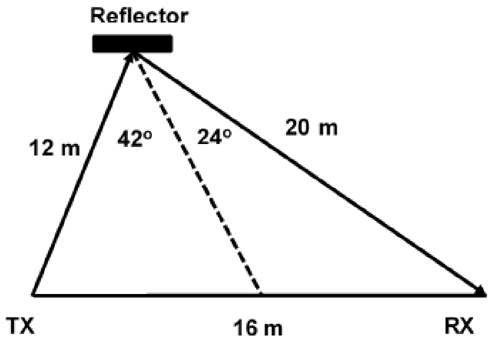


Figure 6 (a). Reflector design of case 2. (b) Throughput performance for case 2.



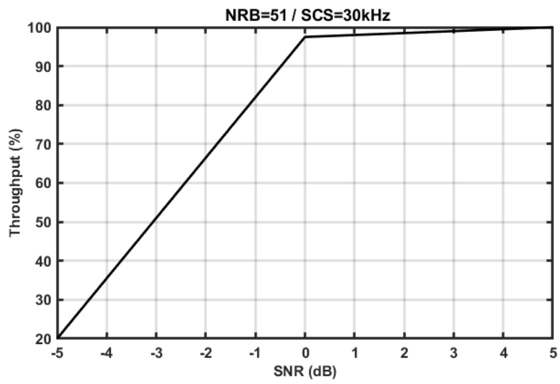
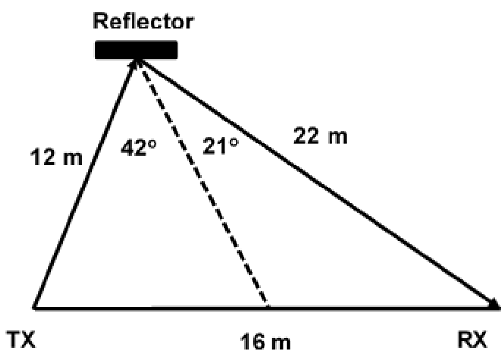


Figure 7. (a) Reflector design of case 3. (b) Throughput performance for case 3.



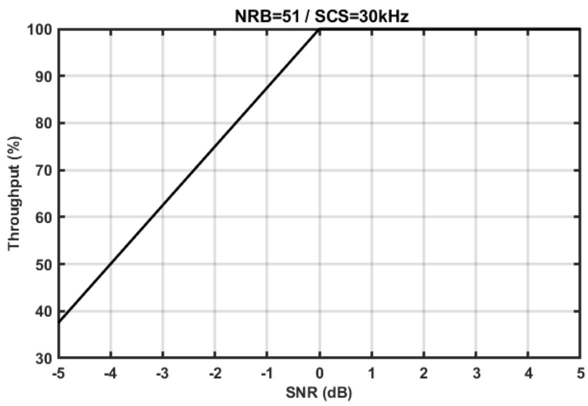


Figure 8 (a). Reflector design of case 4. (b) Throughput performance for case 4.

iv. Case 4, Tx-Rx =16m, Tx-Ref. = 12m, Rx-Ref. = 22m

In this case, the throughput performance was the best among the different cases, as the system achieved 100% of throughput at 0dB and 100% at 5dB, as shown in Figure 8.

v. Case 5, Tx-Rx = 16m, Tx-Ref. = 18m, Rx-Ref. = 8.5m

In this case, the throughput performance was at its worst case, as the system achieved 0% of throughput at all the points, as shown in Figure 9.

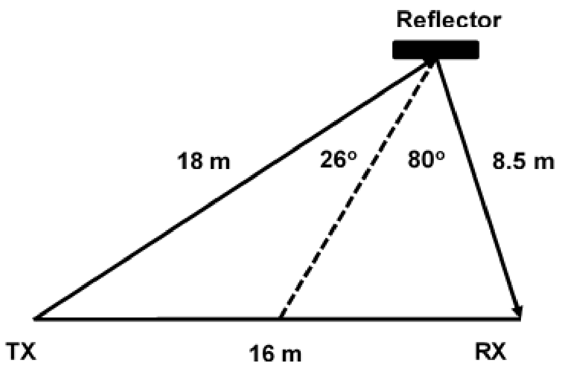
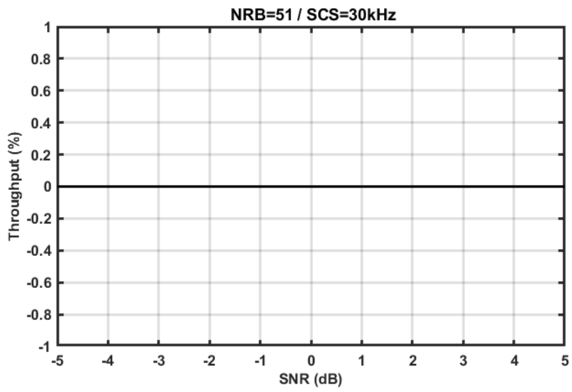


Figure 9 (a). Reflector design of case 5.



(b) Throughput performance for case 5.

Table 4. Different reflector positions moving transmitter or receiver.

	AoA	AoD	Tx-Rx	Tx-Ref.	Rx-Ref.	Throughput % at 0dB	Throughput % at 5dB	Difference at 0dB	Difference at 5dB
No Reflector	0°	0°				42.5	100		
Case 2	30°	22°	7.39	9.23	8.63	100	100	57.5	0
Case 3	10°	30°	6.03	8.12	9.23	80	100	37.5	0
Case 4	40°	35°	10.92	10.44	9.77	40	100	-2.5	0
Case 5	60°	50°	23.4	16	12.44	20	85	-22.5	-15
Case 6	30°	70°	27.32	9.23	22	0	0	-42.5	-100

Comparing the results of the different tested cases, the best case was case 4, where the throughput achieved was 100% when the reflector is spaced 12 meters from the transmitter and 22 meters from the receiver, while the worst case was case 5 when the reflector was spaced 18 meters apart from the transmitter and 8.5 meters apart from the receiver, as shown in Figure 10.

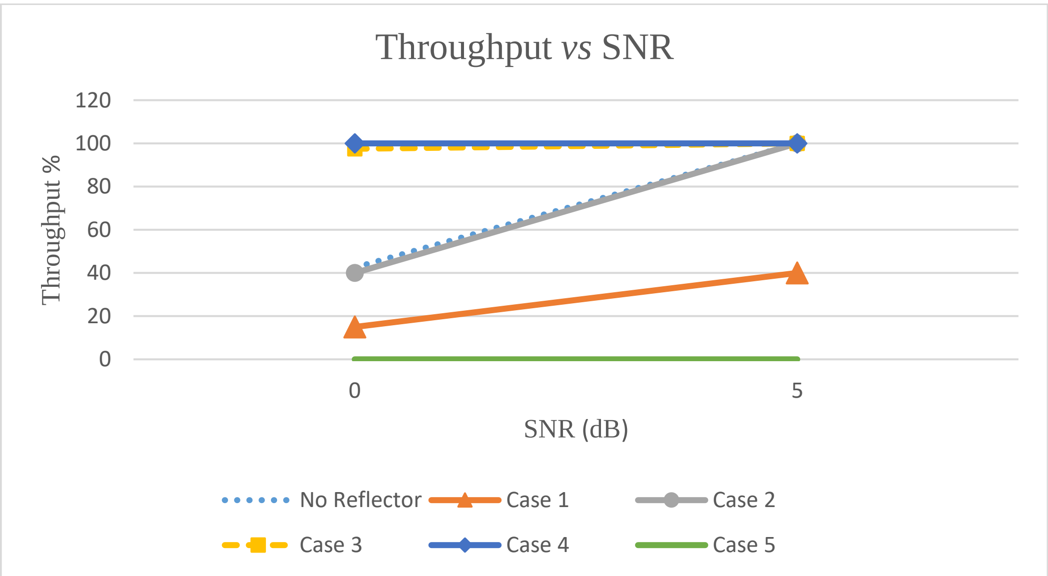


Figure 10. Comparison of different reflector locations with fixed Tx-Rx distance.

b) Moving Transmitter and Receiver

i. Case 1, AoA = AoD = 0° (Default)

For the angle of 0, the throughput percentage reached its maximum at 5 dB and 42.5% of its maximum at 0 dB, as Figure 11 illustrates.

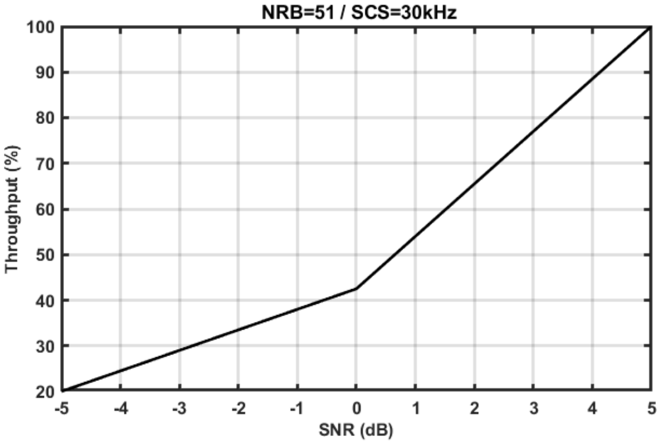
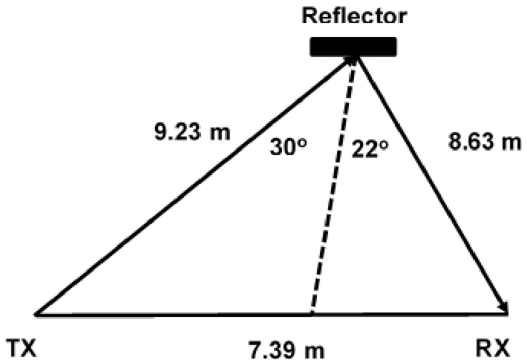


Figure 11. Throughput without reflector.

ii. Case 2, AoA = 30°, AoD = 22°

For this case, the throughput percentage was at its best case which is 100% of throughput all the way from -5 dB to 5 dB (Figure 12 (b)) with an improvement of 57.5% at 0dB compared to the default case (no reflector), with the reflector design shown in Figure 12 (a).



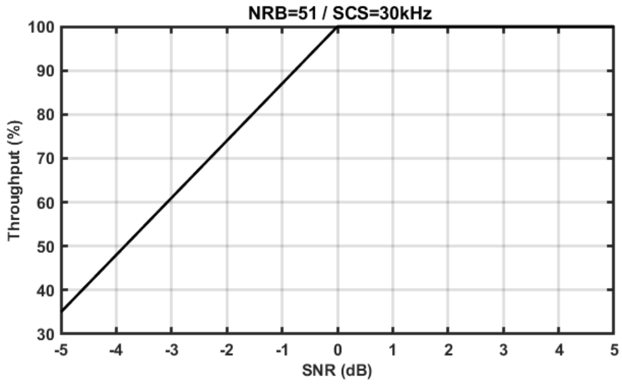
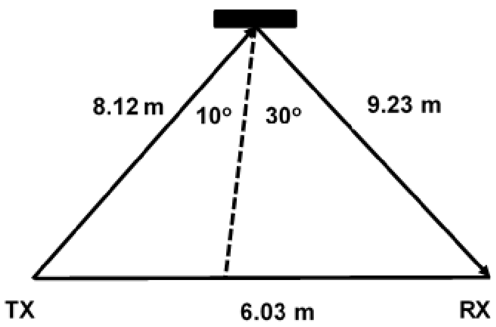


Figure 12 (a). Case 2 reflector design. (b) Throughput for case 2.

iii. Case 3, AoA = 10°, AoD = 30°

For this case, the throughput has improved compared to the default case (no reflector), as the system showed an 80% throughput at 0dB with an improvement of 37.5 % compared to the default case, as shown in Figure 13 (b), while Figure 13 (a) shows the reflector design.



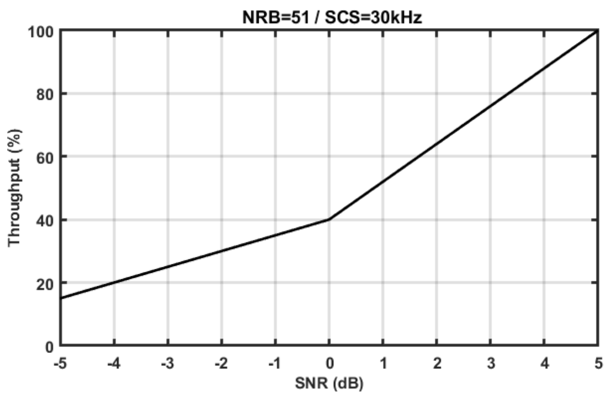
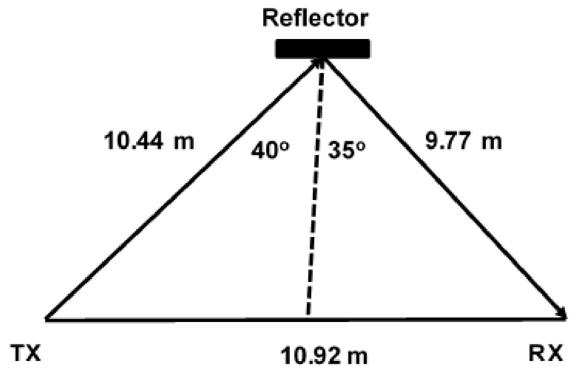


Figure 13 (a). Reflector design of case 3. (b) Throughput for case 3.

iv. Case 4, AoA = 40°, AoD = 35°

This case shows that the performance decreased when using the reflector, as the system showed 40% of throughput at 0dB, which is lower by 2.5% compared to the case of not using the reflector, as shown in Figure 14 (b), while Figure 14 (a) shows the reflector design.



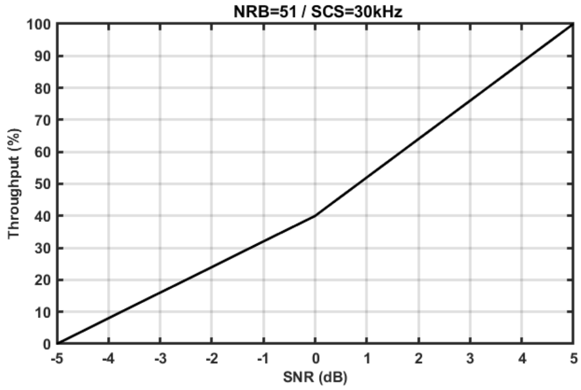
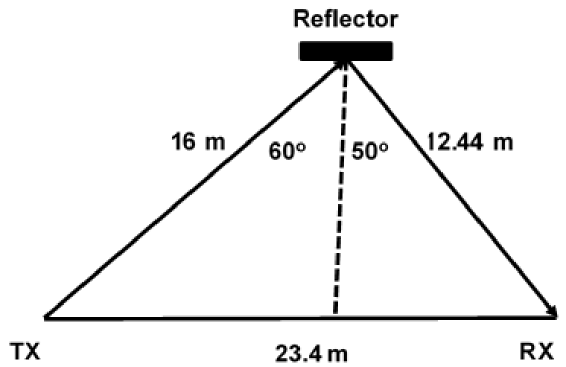


Figure 14 (a). Reflector design of case 4. (b) Throughput for case 4.

v. Case 5, AoA = 60°, AoD = 50

Case 5 shows a significant decrease of throughput percentage compared to the default case (no reflector), with 22.5% decrease at 0dB and 15% decrease at 5dB, as shown in Figure 15 (b).



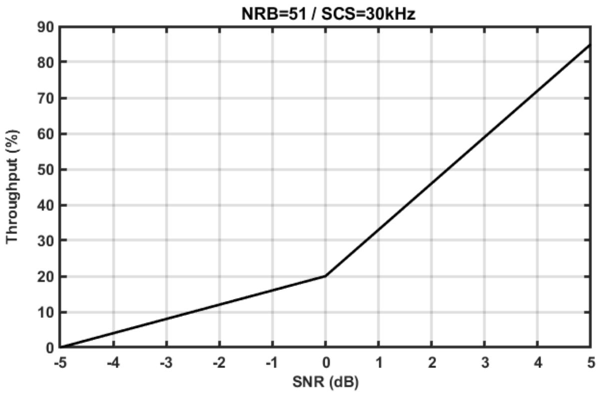
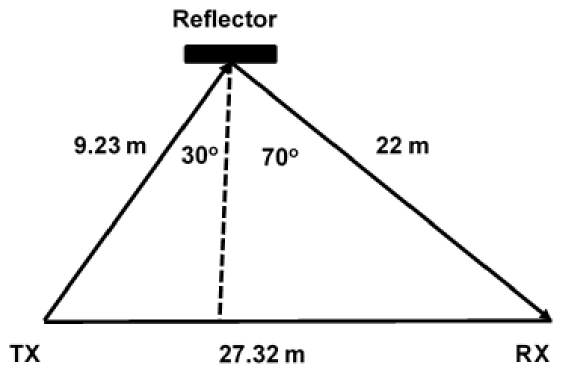


Figure 15 (a). Reflector design of case 5. (b) Throughput for case 5.

vi. Case 6, AoA = 30°, AoD = 70°

The worst case tested in this scenario was when the angle values were 30° for the angle of arrival and 70° and more for the angle of departure, where the throughput resulted in a straight line of 0% for both 0dB and 5dB, which represents a decrease of 42.5% and 100%, respectively, as shown in Figure 16 (b).



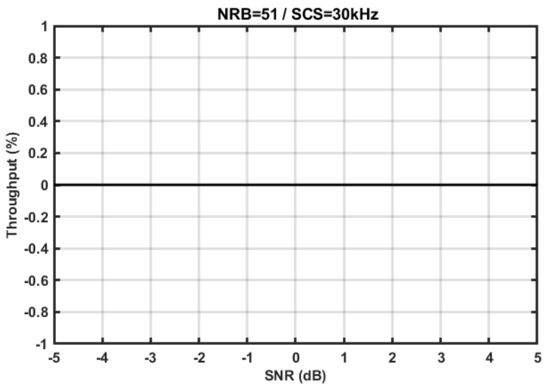


Figure 16 (a). Reflector design of case 6. (b) Throughput for case 6.

As shown in Figure 17, a comparison between the values of throughput in different cases has shown that the throughput performance was at its best in case 2 and at its worst in case 6.

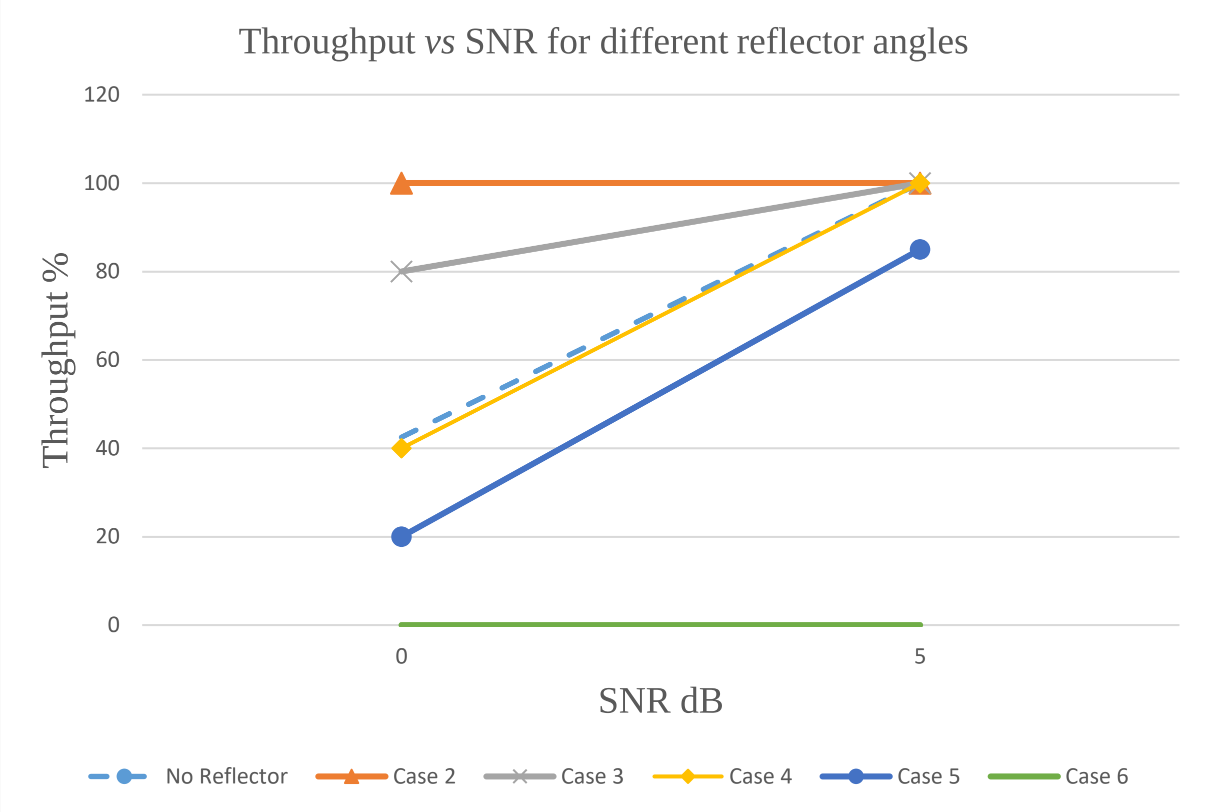


Figure 17. Throughput percentages for different cases of reflector.

4. Conclusion

In this work, an assessment of the throughput performance has been provided based on studying some parameters that might affect it either directly or indirectly. The factors studied were subcarrier spacing, modulation technique in the default CDL channel and the existence of a reflector with two different scenarios. To conclude, it can be argued that using a higher SCS, a greater percentage of throughput can be achieved, whereas for the modulation technique, it is found that higher-order modulation will achieve lower throughput performance and a trade-off should be made to preserve a reasonable spectral efficiency.

Finally, the position of the reflector in case of reflector existence has shown a noticeable effect on the throughput performance for both situations studied; fixed distance and moving users. The results have shown a significant impact of the reflectors in both ways. Further investigation by considering more than one reflector linked with wide coverage milestone can be used in order to improve the performance of system throughput. However, complexity and cost will come into play in this stage. Moreover, investigation of throughput performance using reflectors in mmWave frequency (FR2) in 5G systems could be carried out as well.

References

[1] Tutorialspoint.com, "5G–Challenges," Tutorialspoint Online Education, [Online], Available: https://www.tutorialspoint.com/5g/5g_challenges.

[2] P. Trakadas et al., "Hybrid Clouds for Data-intensive, 5G-enabled IoT Applications: An Overview, Key Issues and Relevant Architecture, " Sensors (MDPI), vol. 19, no. 16, p. 3591, August 2019.

[3] P. K. Gkonis, P. Trakadas and D. I Kaklamani, "A Comprehensive Study on Simulation Techniques for 5G Networks: State-of-the-Art Results, Analysis and Future Challenges," Electronics, vol. 9, no.3, p. 468, March 2020.

[4] C. Nast, "5G Is Coming and It's Fortified with Fiber," [Online], Available: https://www.wired.com/story/5g-is-coming-fortified-with-fiber/.

[5] A. C. Situmorang, D. Gunawan and V. G. Anggraini, "5G Trials on 28 GHz Band in Indonesia," Proc. of the 28th Wireless and Optical Communications Conference (WOCC), pp. 1-5, DOI: 10.1109/WOCC.2019.8770687, Beijing, China, 2019.

[6] W. Khawaja, O. Ozdemir, Y. Yapici, F. Erden and I. Guvenc, "Coverage Enhancement for NLOS mmWave Links Using Passive Reflectors," IEEE Open Journal of the Communications Society, vol. 1, pp. 263-281, 2020, DOI: 10.1109/OJCOMS.2020.2969751, 2019.

[7] O. Ozdemir, F. Erden, I. Guvenc, T. Yekan and T. Zarian, "28 GHz mmWave Channel Measurements: A Comparison of Horn and Phased Array Antennas and Coverage Enhancement Using Passive and Active Repeaters, " arXiv:2002.00121, [Online], Available: https://arxiv.org/abs/2002.00121, February 2020.

[8] ETSI.org, "TS 38.101-1: NR; User Equipment (UE) Radio Transmission and Reception; Part 1: Range 1 Standalone," [Online], Available: https://www.etsi.org/deliver/etsi_ts/138100_138199/13810101/15.03.00_60/ts_13810101v150300p.pdf.

[9] 3gpp.org, "Detailed Configuration of F-OFDM and W-OFDM for LLS Evaluation," 3GPP RAN WG1 #86, Spreadtrum Communications, R1-166999, [Online], Available: https://www.3gpp.org/DynaReport/TDocExMtg--R1-86--31663.htm, August 2016.

[10] 3GPP TS 38.214. NR, "Physical Layer Procedures for Data, Part 5: Physical Downlink Shared Channel Related Procedures (Release 15)," 3gpp Technical Specification Group Radio Access Network, [Online], Available: ttps://www.etsi.org/deliver/etsi_ts/138200_138299/138214/15.02.00_60/ts_138214v150200p.pdf, 2018.

[11] 3GPP TS 38.211. NR, "Physical Channels and Modulation, Part 4: Frame Structure and Physical Resources (Release 15)," 3rd Generation Partnership Project; Technical Specification Group Radio Access Network," [Online], Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/15.02.00_60/ts_138211v150200p.pdf, 2018.

[12] 3GPP TS 38.901, "Study on Channel Model for Frequencies from 0.5 to 100 GHz, Part 7: Channel Model(s) for 0.5-100 GHz (Release 14)," 3rd Generation Partnership Project; Technical Specification Group Radio Access Network, [Online], Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/14.00.00_60/tr_138901v140000p.pdf, 2017.

[13] M. Kerker, Electromagnetic Waves in the Scattering of Light and Other Electromagnetic Radiation, New York, NY, USA: Academic Press, Ch. 2, pp. 8–26, 1969.

[14] S. N. Ghosh, Energy Flow and Boundary Conditions, in Book: Electromagnetic Theory and Wave Propagation, 2nd Ed., New York, NY, USA: CRC Press, Ch. 3, pp. 37–38, 2002.

[15] A. Ghosh and R. Ratasuk, Essentials of LTE and LTE-A, Cambridge, Cambridge University Press, 2011.

[16] S. Le Goff, A. Glavieux and C. Berrou, "Turbo-codes and High Spectral Efficiency Modulation," Proceedings of International Conference on Communications (ICC/SUPERCOMM'94), vol. 2, pp. 645-649, DOI: 10.1109/ICC.1994.368804, New Orleans, LA, USA, 1994.

[17] S. M. Saraireh and A. M. Matarneh,"Higher Level Security Approach for Data Communication System Based on AES Cryptography and DWT Steganography," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 2, no. 3, pp. 179-193, December 2016.

ملخص البحث:

تُنـاقش هـذه الورقـة أداء الاجتيـاز لنظـام راديـوي جديـد مـن أنظمـة الجيـل الخـامس. والهـدف الأساسـي منهـا هـو تطـوير مسـارٍ مـن أجـل تحسـين أداء الاجتيـاز الخـاص بالنظـام عبـر استقصـاء عوامـل معينـة والـتحكّم بهـا. العوامـل التـي خضـعت للدراسـة فـي هـذا البحـث هـي: تقنيـة التعـديل المسـتخدمة، والتباعـد المسـتخدم فـي الحاملـة الفرعيـة فـي قنـاة خـط التـأخير المجمّعـة فـي غيـاب عـاكس، ووجـود عـاكس فـي قنـاة خـط التـأخير المجمّعة.

وبينـت النتـائج أن أداء الاجتيـاز قـد تحسـن مـع زيـادة تباعـد الحاملـة الفرعيـة ومـع اسـتخدام تقنيـة تعـديل ذات رتبـة أقـل. وقـد تـم استقصـاء أثـر كـل مـن وجـود العـاكس وموضـعه فـي أداء الاجتيـاز الخـاص بالنظـام مـع وضـع العـاكس بـين المرسـل والمسـتقبل، بشـيء مـن التفصـيل. وأُخـذت بعـين الاعتبـار حالـة الموضـع الثابـت وحالـة الموضع المتغير للعاكس؛ من أجل الحصول على القيمة المثلى لأداء الاجتياز.

وقـد أسـفرت النتـائج عـن أن وجـود العـاكس يـؤدي الـى الحصـول علـى قيمـة أفضـل لأداء الاجتيـاز مقارنـة بالحالـة التـي لا يُسـتخدم فيهـا عـاكس. وبوجـود عـاكس، وعنـد تباعُـد حاملـة فرعيـة قـدرُهُ (30) كيلـو هيرتـز، يمكـن لأداء الاجتيـاز أن يصـل الـى (100%) عنـد نسـبة إشـارة الـى ضـجيج مقـدارها صـفر ديسـيبل، مقارنـة بــ (40%) عنـد نسـبة إشارة الى ضجيج مقدارها صفر ديسيبل في حالة غياب العاكس.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

فريق دعم هيئة التحرير

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: +962-6-5359949
فاكس: +962-6-7295534

المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)
أيلول ٢٠٢٠ المجلد ٦ العدد ٣

JJCIT

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

مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي