khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Implementation of the Fisher-Yates Shuffle Algorithm in Exam-Problem Randomization on M-Learning Applications

Chandra Kirana*, **Benny Wijaya, Abdul Holil**
Informatics Engineering Study Program
STMIK Atma Luhur
Pangkalpinang, Indonesia
*Chandra.kirana@atmaluhur.ac.id

Abstract-Many schools are currently using conventional approaches in learning material deliveries and examination methods. Conventional examination processes referred to here are the provision of question sheets in paper form. They have several drawbacks, such as students cheating and a waste of paper printing costs. To overcome these problems, we propose an online examination system. The online system leaves students to work on a different question set from other students. The feature is made possible by applying a randomization algorithm. There are several algorithms for scrambling questions, one of which is the Fisher-Yates Shuffle algorithm. This study aims to ease schools in the implementation of quality exams that may find out the level of student understanding of study materials and reduce the risk of cheating. The research product works on Android smartphones, which may be attractive to students and schools. The product allows schools to hold quality exams and reduce paper costs.

Keywords: Fisher-Yates shuffle, exam question, randomization, online exam

Article info: *submitted August 4, 2020, revised December 1, 2020, accepted December 18, 2020*

1. Introduction

The use of information technology has now penetrated all fields, one of which is the field of education. In the world of education, it is necessary to improve the quality, speed, practicality, and convenience in various aspects. One of them is an exam that was previously conventional in nature, to be converted into an online exam. This is an effort to replace the old system with the overall goal of improving the existing system so that it can run more effectively and efficiently. School exams are activities carried out by educational units to measure the achievement of students' competencies, which in turn are used as a measure of achievement of the school[1]. In the world of education, the teacher is one of the spearheads in students' success. Teachers, thus, must be able to follow the development of information technology in order to be more optimal in teaching. The use of technology in making teaching materials and questions will improve the performance of teachers in providing services to students.

This study aims to create an Android-based application that can be used as a practice platform for doing exam questions. Each package of questions given must have a different order of questions from other question packages. This requires scrambling the questions. Many methods can be used to generate random numbers, one of which is the Fisher-Yates Shuffle algorithm which is an algorithm for generating random permutations from a finite set [2]. In addition, the Fisher-Yates Shuffle Algorithm also has a random function in forming a balanced random pattern. This Fisher-Yates Shuffle algorithm is used to randomize the questions in this study.

In general there are many randomization techniques that have been developed by different researchers which can be used in a variety of applications [3]. Among them is a research conducted by Abdi Suhazli who randomized the pieces of a picture in a puzzle game using the fisher yates shuffle method [4]. Research conducted by Widi Aulia Rohmah implements the fisher yates shuffle algorithm to randomize questions in a quiz game, where the questions that appear are randomized and the user does not easily guess the next question [5]. Another research was also conducted by Shafali Agarwal. In that research, the process of making a pixel permutation key was carried out using the knuth shuffle method and dynamic diffusion from random images [6]. Fyan Dimas Pratama's research in 2019 regarding the application of online daily tests using the Linear Congruential Generator (LCG) method concluded that this method can reduce question leakage [7]. A quite interesting research was also conducted by

Imam Haditama, Cepy Slamet and Deny Fauzy Rahman in 2016 regarding the Implementation of the Fisher-Yates and Fuzzy Tsukamoto Algorithm in an Android-Based Sundanese Tone Guess Quiz Game. The results of his research showed that the Fisher-Yates algorithm is able to determine the solution of non-multiple randomization and multiple object randomization [8]

In this study, the authors used the fisher yates shuffle algorithm for the randomization process because the algorithm uses a shuffling technique used to change the order of an element randomly where the randomization process is unbiased and random [9]. Though the algorithm has a dynamic shuffling nature, the modern form of the algorithm is more efficient. The implementation enhances the time complexity to $O(n)$, from $O(n^2)$. Here, the rolled number indicates the position of the element removed from the current list and inserted into the shuffled list and number at the end of the current list is placed at the same position that yields further shuffling at each iteration. This makes the system more dynamic. When the list contains one element, it is removed and inserted into shuffled list [10].

2. Method

In this study, the authors created a framework to facilitate the development. This framework can be seen in Figure 1.

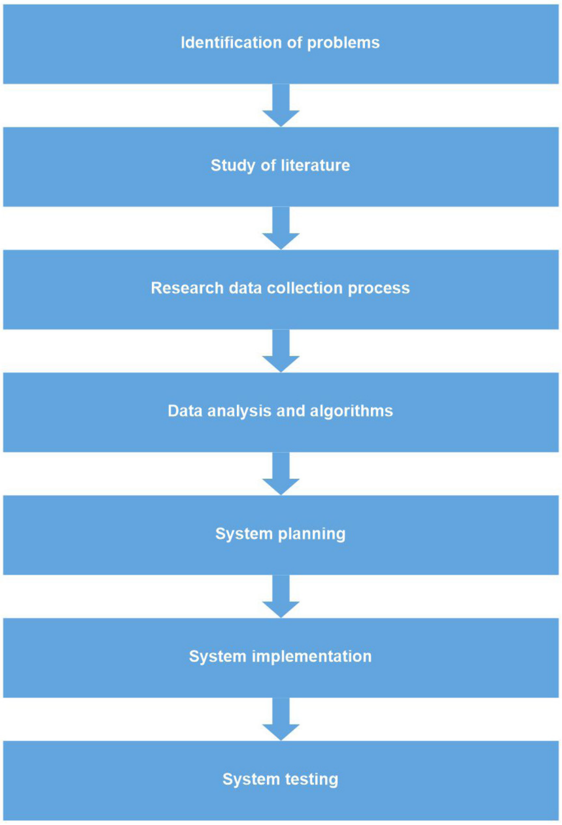

Identification of problems
Study of literature
Research data collection process
Data analysis and algorithms
System planning
System implementation
System testing

Figure 1. Research Framework

The description is as follows

1. Problem Identification
   The process carried out at this stage is to formulate the problems that occur today and which are the objects in the research. The formulated problem is related to the process of randomizing questions in the implementation of the exam and testing the Fisher-Yates Shuffle algorithm in randomizing questions..
2. Literature Review
   This stage is then the authors perform data analysis on the problems that occur. One of the analysis processes carried out by the author is by studying the literature related to the problem tracking algorithm. The data source is obtained by the author through journals and books related to question randomization and the Fisher-yates shuffle algorithm.
3. Data collection
   At this stage the writer carried out the data collection process where the writer took the data sample by means of interviews, observations, and observations.
4. Data analysis
   At this stage, the authors carry out the process of analyzing the data needed in this study, such as data for the randomization process and data related to the Fisher-yates shuffle algorithm..
5. System planning
   At this stage the authors carry out the system design process to be built including the desired form of system design and planning in making the system. The purpose of this design is to facilitate the next process, namely at the stage of making the desired system.
6. System Implementation
   After the system design process is carried out, the next step is to build the desired system.
7. System Testing
   This stage is where the writer ensures whether the process previously carried out is running as expected or not.

3. Result

a. Data analysis

The analysis in this study includes how the fisher-yates shuffle algorithm works in the process of randomizing the questions that are applied in the online exam system that has been built. An overview of the system can be seen in Figure 2.

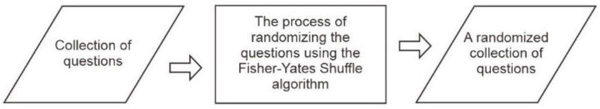

Collection of questions
The process of randomizing the questions using the Fisher-Yates Shuffle algorithm
A randomized collection of questions

Figure 2. System Overview

b. Analysis of the Fisher-Yates Shuffle Algorithm

This development is one way that can be done to minimize cheating in doing practice questions. The process of implementing the algorithm can be seen in Figure 3.

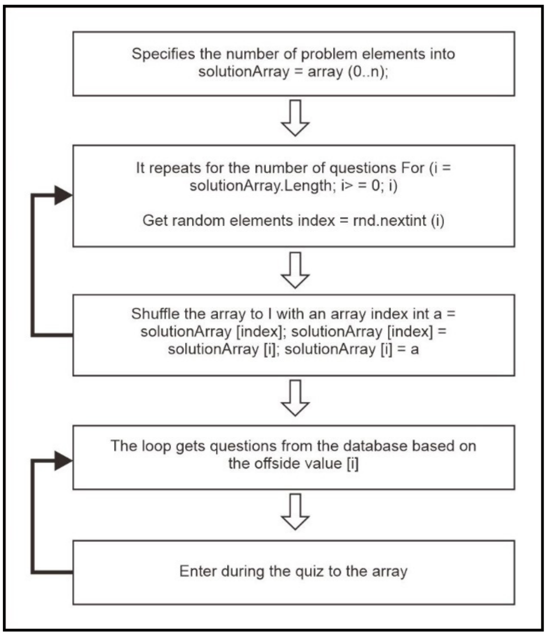

Specifies the number of problem elements into solutionArray = array (0..n);
It repeats for the number of questions For (i = solutionArray.Length; i> = 0; i)
Get random elements index = rnd.nextint (i)
Shuffle the array to I with an array index int a = solutionArray [index]; solutionArray [index] = solutionArray [i]; solutionArray [i] = a
The loop gets questions from the database based on the offside value [i]
Enter during the quiz to the array

Figure 3. Fisher-Yates Shuffle Algorithm Process Flow

The steps used to generate a random permutation for problems 1 to *N* can be seen as follows [3]:

1. Write down the questions from number 1 to number *N*.
2. Choose a random question *K* between 1 and the number of questions that have not been crossed out.
3. Count from the bottom position. Cross out the *K* questions that have not been crossed out and write the questions in another place.
4. Repeat step 2 and step 3 until all the questions have been crossed out.
5. The order of the questions written in step 3 is the random permutation of the initial problem.

Table 1. Randomized questions using the Fisher-Yates Algorithm

Range	Roll	Scratch	Result
		1,2,3,4,5,6,7,8,9,10,11, 12,13,14,15	
1-15	6	1,2,3,4,5,7,8,9,10,11,12,13, 14,15	6
1-14	10	1,2,3,4,5,7,8,9,10,12,13, 14,15	11,6
1-13	3	1,24,5,7,8,9,10,12,13,14, 15	3,11,6
1-12	1	2,4,5,7,8,9,10,12,13,14,15	1,3,11,6
1-11	6	2,4,5,7,8,10,12,13,14,15	9,1,3,11,6
1-10	4	2,4,5,8,10,12,13,14,15	7,9,1,3,11,6
1-9	7	2,4,5,8,10,12,14,15	13,7,9,1,3,11,6
1-8	7	2,4,5,8,10,12,15	14,13,7,9,1,3,11,6
1-7	2	2,5,8,10,12,15	4,14,13,7,9,1,3,11,6
1-6	4	2,5,8,12,15	10,4,14,13,7,9,1,3,11,6
1-5	5	2,5,8,12	15,10,4,14,13,7,9,1,3,11, 6
1-4	4	2,5,8	12,15,10,4,14,13,7,9,1,3, 11,6
1-3	3	2,5	8,12,15,10,4,14,13,7,9,1, 3,11,6
1-2	2	2	5,8,12,15,10,4,14,13,7,9, 1,3,11,6
			2,5,8,12,15,10,4,14,13,7, 9,1,3,11,6

The next step is to enter the question attribute into scratch (a list of questions that have not been selected) then create a range (the number of questions that have not been selected). Next do the randomization process, and then see the roll (for a question that is selected from all the existing questions). After that, the results of the questions that have been selected are entered into the results (the results of all the questions that have been randomized) [2]. The process of implementing the randomization algorithm for fifteen questions is exemplified in Table 1.

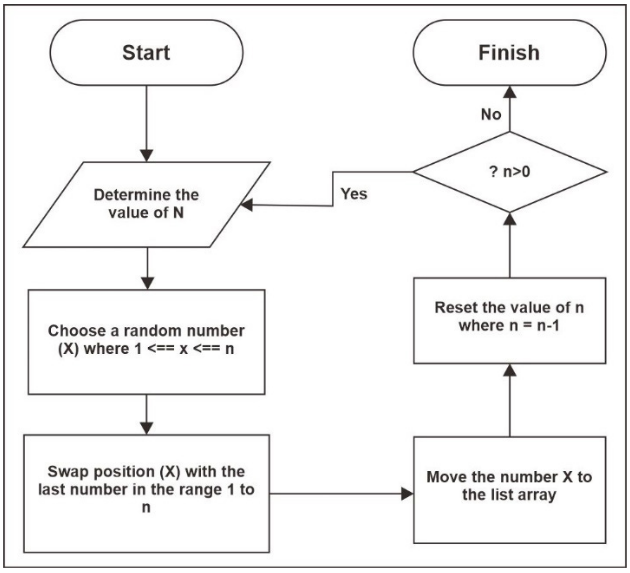

Start
Finish
No
? n>0
Yes
Determine the value of N
Choose a random number (X) where 1 <== x <== n
Reset the value of n where n = n-1
Swap position (X) with the last number in the range 1 to n
Move the number X to the list array

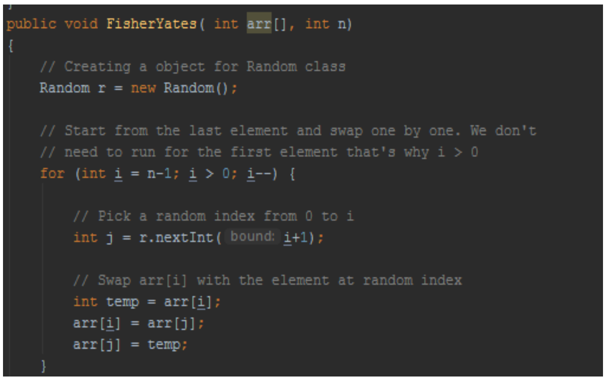
Figure 4. The fhiser yates shuffle randomization method [3]

From Figure 4, it can be seen that the randomization is complete when the entire array has been scrambled. The fisher-yates shuffle randomization method produces a randomized array sequence.

c. System Testing

This research resulted in the implementation of the fisher yates shuffle algorithm which is used for the randomization process in mobile-based M-Learning applications. Algorithm testing is carried out on practice exam questions, thus each student will get a different order of questions from other students.

Algorithm implementation into the M-Learning application can be seen in Figures 5 and 6.



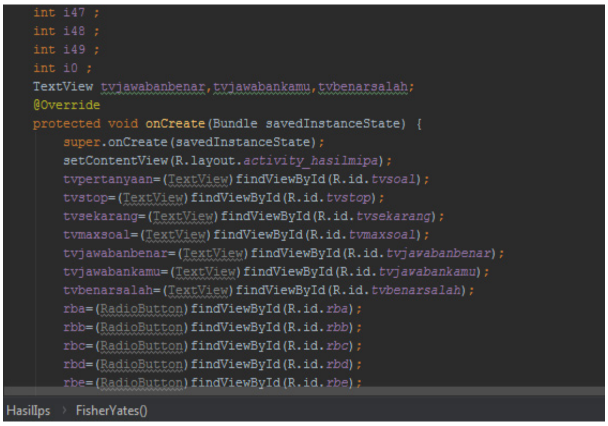
```
public void FisherYates( int arr[], int n)
{
    // Creating a object for Random class
    Random r = new Random();

    // Start from the last element and swap one by one. We don't
    // need to run for the first element that's why i > 0
    for (int i = n-1; i > 0; i--) {

        // Pick a random index from 0 to i
        int j = r.nextInt( bound: i+1);

        // Swap arr[i] with the element at random index
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
```

Figure 5. The process of selecting Random Problem



```
    int i47 ;
    int i48 ;
    int i49 ;
    int i0 ;
    TextView tvjawabanbenar,tvjawabankamu,tvbenarsalah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hasilmipa);
        tvpertanyaan=(TextView)findViewById(R.id.tvsoal);
        tvstop=(TextView)findViewById(R.id.tvstop);
        tvsekarang=(TextView)findViewById(R.id.tvsekarang);
        tvmaxsoal=(TextView)findViewById(R.id.tvmaxsoal);
        tvjawabanbenar=(TextView)findViewById(R.id.tvjawabanbenar);
        tvjawabankamu=(TextView)findViewById(R.id.tvjawabankamu);
        tvbenarsalah=(TextView)findViewById(R.id.tvbenarsalah);
        rba=(RadioButton)findViewById(R.id.rba);
        rbb=(RadioButton)findViewById(R.id.rbb);
        rbc=(RadioButton)findViewById(R.id.rbc);
        rbd=(RadioButton)findViewById(R.id.rbd);
        rbe=(RadioButton)findViewById(R.id.rbe);
```

Figure 6. The process of displaying the results of scrambling questions

The results of the randomization test with the Fisher Yates shuffle algorithm can also be seen in Table 2.

Table 3. Result of randomization test

Testing to-	The result of the appearance of the question number
1	37,16,39,11,7,47,10,45,31,9,13,36,35,26,4,38,28,46,27,14,19,12,6,41,48,42,15,29,30,17,1,22,40,5,32,8,20,43,24,50,33,18,2,21,23,3,44,25,34,49
2	21,30,34,26,36,24,39,8,17,29,25,3,14,42,49,47,13,32,50,1,27,35,28,16,48,10,23,15,2,7,41,33,4,46,5,38,9,18,19,45,43,28,44,12,6,11,22,40,37,20
3	18,16,20,6,38,36,39,40,7,29,32,30,31,44,13,8,14,37,27,24,2,42.9,12,46,19,23,26,11,35,17,48,33,28,22,43,15,49,41,21,10,3,47,50,44,1,25,5,34,45
4	5,21,27,28,42,24,47,1,19,37,25,22,43,29,18,39,35,6,13,49,3,16,11,40,14,48,41,32,46,15,12,30,34,17,38,31,20,36,45,2,7,50,10,44,4,9,8,23,33,26
5	33,3,48,49,14,30,6,40,31,39,46,36,16,18,10,34,28,29,25,8,19,22,23,38,4,9,32,26,43,41,20,2,24,12,47,5,37,11,21,50,17,1,7,35,45,15,27,13,42,44

In Figure 7, it can be seen that the fisher yates shuffle algorithm was successfully applied to the M-Learning application by using several cellphones simultaneously with different questions.

Tests in this study used two tests, namely alpha and beta testing. Alpha testing is also known as blackbox testing. This test is carried out using software specifications without referring to procedures from the internal system being made. The purpose of this test is to test whether the components in the system are in accordance with the desired design. This blackbox test only looks at the basics of the system and ensures that the input made can be accepted by the system properly and produces output as expected. The testing process can be seen in Table 3.

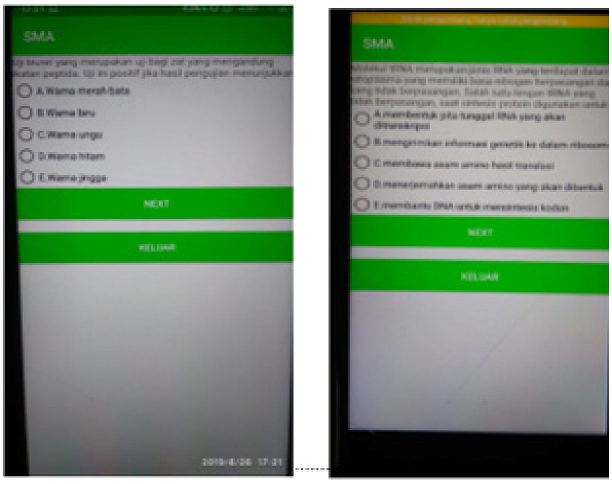


Figure 7. Algorithm testing on the M-Learning application

Table 3. Black Box Test Results

Activity	Output	Result
Login	The system displays the main page	According to the design
List	The system displays a list form and stores data into the database	According to the design
Select Material	The system displays the Material page	According to the design
Select Problem	The system displays the Questions page	According to the design

Based on Table 3, it can be seen that all system functions that are built can run well using alpha testing (blackbox). If in system use an error occurs it is caused by a system user who does not provide the correct input. In addition to Alpha testing, this study also uses Beta testing, namely testing using a questionnaire. The test sample is as many as 20 individuals, consisting of 15 students and 5 teachers. The list of questionnaire results from this test can be seen in Table 4.

Table 4. Beta Testing Results with a Questionnaire

Questionnaire	Answer (%)			
	1	2	3	4
Do you agree, if the system that is made has a good user interface?	0%	0%	0,45%	0,55%
Do you agree, if the system created can make it easier for the school in the learning process, especially the exam process?	0%	0%	0,3%	0,7%
Do you agree, if this system makes it easy to use?	0%	0%	0,25%	0,75%

Questionnaire	Answer (%)			
	1	2	3	4
Does Ands agree, if the system created can randomize the questions, so that the questions differ from one another?	0%	0%	0,1%	0,9%

Note: 1: Strongly disagree; 2: Disagree; 3: Agree; 4: Totally agree

The results of this Beta test can be concluded as follows: (1) The system created has a good user interface and is easy to use; (2) Users can follow and use the system in accordance with the instructions provided; (3) This system is able to provide different questions from one another by applying the Fisher-Yates Shuffle algorithm as a process of randomizing the questions..

4. Conclusion

This M-Learning system has been built with the Android operating system. While the development of the system uses a prototype model and development tools using UML. For testing, this study uses Beta and Alpha testing. The results of Alpha testing using the Black box show that all functions in the M-Learning system can work properly in accordance with the system design, especially for the process of randomizing questions using the Fisher-Yates Shuffle algorithm. Beta test results can be concluded as follows: (1) The system created has a good user interface and is easy to use; (2) Users can follow and use the system in accordance with the instructions provided; (3) This system is able to provide different questions from one another by applying the Fisher-Yates Shuffle algorithm as a process of randomizing the questions. Presentation of the results of this system using mobile technology, which is based on Android, so that the system used can provide convenience in its use and where and whenever students can carry out the learning process without having to go to school.

Reference

[1] Gunawan And D. A. Prabowo, "Sistem Ujian Online Seleksi Penerimaan Mahasiswa Baru Dengan Pengacakan Soal Menggunakan Linear Congruent Method (Studi Kasus Di Universitas Muhammadiyah Bengkulu)," *Inform. Upgris*, Vol. 3, No. 2, Pp. 143–151, 2017.

[2] M. A. Hasan, Supriadi, and Zamzami, "Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Lancang Kuning Riau)," *J. Nas. Teknol. dan Sist. Inf.*, vol. 03, no. 02, pp. 291–298, 2017.

[3] S. A., S. N., and M. N., "Generating Random Data using 3 Nonlinear Functions," *Int. J. Comput. Appl.*, vol. 152, no. 4, pp. 6–10, 2016, doi: 10.5120/ijca2016911776.

[4] A. Suhazli, A. Atthariq, and A. Anwar, "Game Puzzle 'Numbers in English'Berbasis Android Dengan Metode Fisher Yates Shuffle Sebagai Pengacak Potongan Gambar," *J. Infomedia*, vol. 2, no. 1, pp. 1–6, 2017, doi: 10.30811/.v2i1.476.

[5] W. A. Rohmah, A. Asriyanik, and W. Apriyandari, "Implementation of the Algorithm Fisher Yates Shuffle on Game Quiz Environment," *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 161–172, 2020, doi: 10.31289/jite.v4i1.3863.

[6] S. Agarwal, "A fractal based image cipher using Knuth shuffle method and dynamic diffusion," *Int. J. Comput. Networks Commun.*, vol. 11, no. 4, pp. 81–100, 2019, doi: 10.5121/ijcnc.2019.11405.

[7] F. D. Pratama, "APLIKASI ULANGAN HARIAN ONLINE Menggunakan Metode Linear Congruential Generator Berbasis Website," Universitas Teknologi Yogyakarta, 2019.

[8] I. Haditama, C. Slamet, and D. F. Rahman, "IMPLEMENTASI Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android," Vol. I, No. 1, Pp. 51–58, 2016.

[9] M. Tayel, G. Dawood, and H. Shawky, "Block cipher S-box modification based on fisher-yates shuffle and ikeda map," *Int. Conf. Commun. Technol. Proceedings, ICCT*, vol. 2019-October, pp. 59–64, 2019, doi: 10.1109/ICCT.2018.8600161.

[10] T. K. Hazra, R. Ghosh, S. Kumar, S. Dutta, and A. K. Chakraborty, "File encryption using Fisher-Yates Shuffle," *2015 Int. Conf. Work. Comput. Commun. IEMCON 2015*, no. March 2016, 2015, doi: 10.1109/IEMCON.2015.7344521.