Forward and Backward Random LSBs Steganography Against Visual Attacks

Eman A. Abu Tuema[1], Tawfiq Barhoom[2]
Faculty of Information Technology, Islamic University of Gaza, Gaza, Palestine

Abstract

Steganography is the field of science that deals with how to hide secret data (text, image, audio,etc.) into another media objects as a cover or carrier in a way that no one can notice that there is hidden data. There are many techniques for applying steganography. The most popular technique is Least Significant Bit (LSB). This technique, however, is easy to be attacked and hidden data can be retrieved in the case of sequential-based hiding. This research introduces a new randomized Forward-and-Backward LSB algorithm for image steganography differs from other algorithms in the way of hiding secret data. We developed an algorithm based on two indicators: one for determining the cover bytes and the other for specifying the cover bytes capacity. Random Forward-and-Backward selection of cover bytes makes the proposed algorithm robust against visual attacks. Moreover, the number of bits to be hidden using the proposed algorithm is not fixed; hence increasing the capacity of the cover bytes (payload). The proposed algorithm was tested, evaluated and compared with existing algorithms. Our proposed algorithm achieved better results than other methods with respect to steganography aspects: imperceptibility, capacity (i.e., payload) and robustness (resistance to attacks); especially against visual attack.

1. Introduction

Steganography is the process of hiding data in the media object such as text, image, audio or video and the secret data may also be text, image, audio, etc in such a way that others will not be able to notice [1]. Steganography is an information security field that deals with how to carry data with protection from unauthorized individuals or systems. Unlike cryptography which differs in the way of protecting the data, steganography prevents the discovery of existence of communication with no changes in the structure of secret data to be hidden. However, cryptography prevents unauthorized persons from discovering the contents of communication by converting secret data to an understandable form before carrying it [2]. Both techniques could be combined together in order to achieve more protection for secret data.

There are many different types of steganography depending on the cover object used to carry secret data:

- *Image Steganography*: Cover objects are images whose pixels are used for hiding the secret data.
- *Network Steganography*: Network protocols such as IP, TCP and UDP are used to hide secret data [3].
- *Text Steganography*: Cover objects are texts where the number of tabs, white spaces, uppercase letters, mouse coding are used to achieve the information hiding.
- *Video Steganography*: The process of hiding secret data inside video files [4].
- *Audio Steganography*: Hiding some text or audio information inside host audio files [4].

Image steganography is the most popular technique that is used to hide secret data. The cover media is an image whose contents are pixels and each pixel is represented by one byte; a stream of eight bits as in the gray scale image model, or represented as a mixture of three bytes; one for each color channel in Red, Green and Blue (RGB) image model. The mage which carries secret data is called 'stego' image. Figure 1 illustrates the process of image steganography.

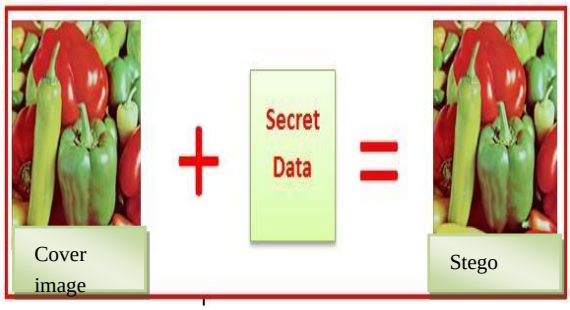


Figure 1. Image steganography process

There are many techniques in use to hide data in cover images, and the most popular one is LSB based technique because it is easy to use and easy to develop algorithms. Existing LSB-based algorithms can be easily enhanced by just hiding data in the last right bit (i.e., LSB) of cover byte.

2. LSB Hiding Technique

The most common and popular technique for hiding secret data is by replacing the secret bits with the LSBs of carrier bytes. The following example shows how to hide eight secret data bits by replacing the eight LSB bits of eight cover bytes, as shown in Table 1.
The three marked bits by parenthesis in Table 1 are only changed 'altered' after inserting one secret byte inside eight cover bytes using the LSB hiding technique.
There are many advantages of the LSB technique [5].

- Easy to understand and use.
- Does not affect the cover media size, it just replaces bits with the least important bits.
- The produced stego image is similar to the real carrier image.

Table 1. LSB process for hiding data

Secret data byte to be hidden is [10001101]		
Cover bytes before the LSB-based hiding		
00111011	10110110	11001101
01001101	10100101	10111100
11001101	10101010	01010110
Cover bytes after the LSB-based hiding		
00111011	10110110	11001101
01001101	1010010(0)	10111100
1100110(0)	1010101(1)	01010110

- Does not produce noticeable changes to the cover data and this is based on two factors:
  - Change is made on the last right bit which is the least important bit in the cover byte.
  - Last right bit similar to the secret bit is left unchanged as shown above in Table 1.

LSB technique, however, has some disadvantages. It is quite easy to tell if an image has been steganographed with an enhanced LSB attack. Also, it consumes too many cover bytes to hide few bytes of data.

3. Visual Attack

Visual attack is the simplest form of steganalysis. The stego image can be scanned with the naked eye to see if there is hidden data in it [**18**]. Visual attacks occur because secret hidden messages can be seen on the low bit planes of an image as they overwrite visual structures. This usually happens in bit map images (BMP) and it is based on bit plane of the image [6]. There are three factors for successful visual attack:

1. The message should be hidden in sequential–based form.
2. The length of secret message is less than the maximum size of the bit plane.
3. The secret message is not encrypted, because the encryption process reduces the chance of success.

This paper introduces a new forward-and backward randomized LSB algorithm for image steganography that differs from other algorithms in the way of hiding data. Our forward-and-backward algorithm is based on two indicators: determining cover byte and determining cover byte capacity. The aim of the proposed algorithm is to achieve the factors of the strongest steganography system.

4. Related Work

Many steganographic algorithms were implemented with various degrees of strength and weakness [7]. In [1], the authors introduced an overview about steganography techniques and their classification. They presented an in-depth look of steganography concepts, history and the most available techniques.

The authors in [8] proposed an algorithm that XORs secret key with the value of red channel of cover image in order to determine the position where to hide secret data. If the result of the XORing is 0, the hiding is done in the blue channel, otherwise it is done in the green channel. Implementing this algorithm, however, limits the capacity of cover bytes.

Image steganography was implemented with DES encryption approach in the work reported in [9]. Information to be hidden were first encrypted by DES encryption before applying the image LSB steganography. The results showed that the encryption algorithm enhanced the anti-detection of the image steganography.

In addition, many studies have been conducted for random image steganography some of which were indicator-based. In [10], the researchers presented pixel indicator technique (PIT) for RGB images. Their technique used the two LSBs of one channel as indicator to hide one or two bits in the other two channels. The indication channel is changed from pixel to another randomly and that increased security and capacity.

Also, the authors in [11] introduced a different indicators-based algorithm. In their algorithm, they used two indicators: indicator for selecting the cover byte for where to hide data and indicator for

determining the number of bits to be hidden in the cover byte. Using secret key and random determination of cover byte, they increased security and capacity. This algorithm, however, works only forward, i.e., from the beginning of cover image.

LSB technique implemented using RC4 encryption algorithm with stego key was introduced in the work reported in [12]. Cover bytes were selected randomly to increase security. Stego image quality was improved with the use of inversion byte to hide massages.

Another indicator-based algorithm was proposed in [5]. The algorithm, called ST_Rindicator steganography, used benchmark RGB image (with png and bmp extension) as a cover media where each pixel is represented by three bytes (24 bits) red, green, and blue. The process of hiding depended on the pixel indicator technique; the Rindicator. The authors used the same principle of the LSB where the secret message is hidden in the LSBs of the pixels, with more randomization in choosing the number of bits and the color channels that are used. In addition, secret bits may be embedded into one or two bits. Use of randomization made the method robust against steganalysis and it increased the capacity of information.

The researchers in [13] proposed steganography system composed of two parts. In the first part, they used AES cryptography algorithm to cipher the secret message's bits. The second part was for hiding cypher secret bits by using the MSBs bits of cover bytes as indicators. If at least two MSBs of the RGB channel's value is 1, it is light, otherwise it is dark. By using AES and random selection of where to hide, the authors increased the imperceptibility and robustness.

An optimized image steganography approach was reported in [14]. Their study consisted of three phases. In the first phase, they hide the secret data by XORing it with the LSB of cover bytes. In the second phase, they used Genetic Algorithm (GA) in a heuristic approach to find best solution to optimize the stego image. The last phase was for extracting the secret data.

The previous studies focused on one or two of the three sides of the steganography triangle: imperceptibility, robustness and capacity. Hiding data either sequential or random start from the beginning to the end of the cover medium (forward). This results in easier recovery for visual attacks. In the proposed algorithm, however, the process of hiding data is random and therefore is difficult to retrieve. Our work also focused on all three sides of the triangle in varying degrees. Moreover, the hiding process is expanded to be performed from both directions (forward and backward).

5. Randomized-Based LSB Algorithm

In our proposed algorithm for embedding process, the secret data bits were embedded in LSBs of the cover images depending on an indicator. An indicator is any bit of the cover byte other than the two LSBs used to hide the data. Here, our indicator is used for determining the number of secret bits to be hidden into every cover byte. A "capacity" indicator, Ci, is the MSB of the cover byte. If the value of the MSB is 0, only one bit is embedded in the cover byte, otherwise two bits are embedded. For determining the cover byte where secret bits are hidden into, a "where" indicator, Wi, was used in the algorithm which scans the image in eight cycles.
The first four cycles scan forward the even bytes from the beginning to the end of the cover image. These cycles are termed eC1, eC2, eC3 and eC4.Table 2 lists the four cycles:

Table 2. Even cycles for the hiding process

Cycle no.	The two MSBs	No. of secret bits to be hidden
eC1	00	1 bit
eC2	01	1 bit
eC3	10	2 bits
eC4	11	2 bits

The same process is repeated for the odd bytes but in backward manner (from the end of image to its beginning). The odd cycles are termed oC1, oC2, oC3 and oC4.

Using Wi increases the robustness of the hiding process. Therefore, this algorithm makes the process of retrieving the hidden data more complex and therefore increases the security against visual attacks. Using Ci, on the other hand, increases the capacity over normal LSB-based algorithms that hide just one bit in the LSB of the cover bytes. Figure 2 below illustrates the process of hiding data using the proposed algorithm. The retrieval process is the inverse of the hiding process as illustrated in Figure 3.

6. Experimental Results and Discussion

The proposed algorithm was tested using dataset images collected from USC-SIPI-ID [17] a well-known dataset researcher use to test their algorithms. These images were Pepper, Airplane, Baboon, Boat and House. Two other images were obtained from the Internet: Splash and Friend. As spatial domain is used for the hiding process, all data set images were of the type PNG and BMP of size 512×512 pixels. In

order to test the proposed algorithm, experiments were conducted using cover bytes at varying rates from 10% to 50% of the cover image size with 5% increment in each experiment. Thus the size of hidden data increases in every experiment. The bit rates per pixel (bpp) of the hidden secret data were 1.0, 1.5, 2.0 and 2.8.

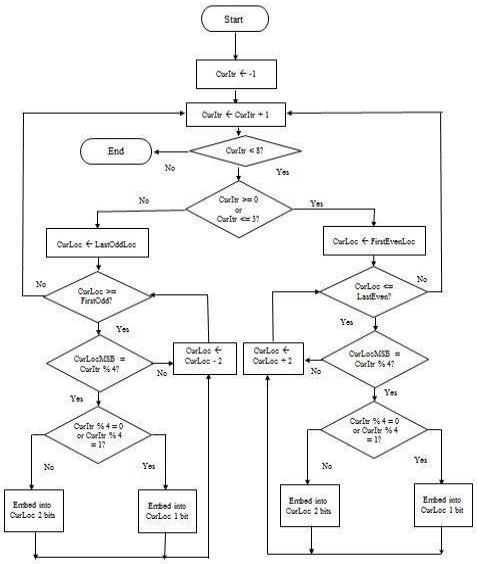


Figure 2. Forward-and-backward hiding process flowchart

(*CurItr= Current Iteration, CurLoc= Current Location, CurLocMSB= Current Location MSB)*

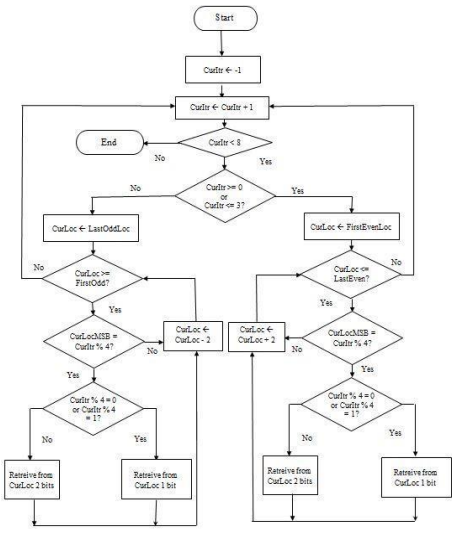


Figure 3. Forward-and-backward rettreiving process flowchart

Mean Square Error (MSE) and Peak-toSignal-Noise-Ratio (PSNR) were calculated for all the stego images using equations (1) and (2) as follows**.**

$$MSE = \frac{\sum_{MN} [I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (1)$$

where *M* and *N* are the number of rows and number of columns of the image, respectively.

$$PSNR = 10 * Log^{10} \left(\frac{R^2}{MSE} \right)$$

(2) where *R* is the maximum fluctuation in the input image data type [15].

The average MSE and average PSNR for stego images were calculated in order to measure the imperceptibility and find the impact of the hiding process. The results are displayed in Table 2. The MSE is the statistical difference between cover and stego images as illustrated in Table 3 and Figure 4. It can be noticed from the MSE values in Table 3 that the average MSE is between 0.23 and 0.46. These averages are relatively small which indicate small difference between the images (stego and cover).

Table 3. MSE and PSNR between cover and stego images

No.	Image name	Average MSE	Average PSNR
1	Pepper	0.25	55.43
2	Splash	0.23	55.77
3	Airplane	0.45	51.66
4	Friend	0.46	52.76
5	House	0.29	54.86

Likewise, PSNR rate was used for measuring the similarity between the cover image and all of its stego images. The average PSNR was between 55.77 and 52.76 as listed in Table 3 and shown in Figure 5. These figures indicate that the proposed algorithm is very imperceptible.

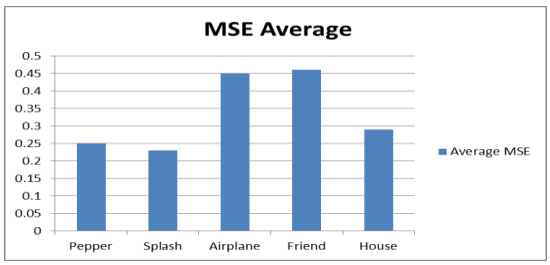


Figure 4: MSE averages for cover and stego images

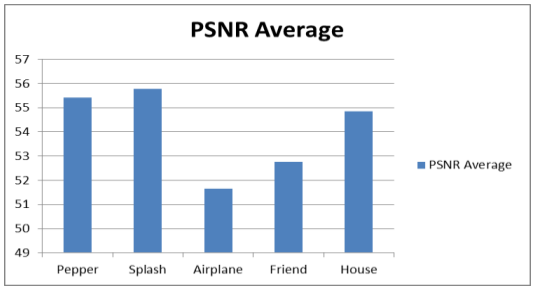


Figure 5. PSNR averages for cover and stego images

For each cover image there were nine stego images with secret data of different sizes. These stego images were subjected to the steganalysis tool, called StegExpose, in order to test the proposed algorithm's robustness and detectability. Table 4 shows the results of using the tool.

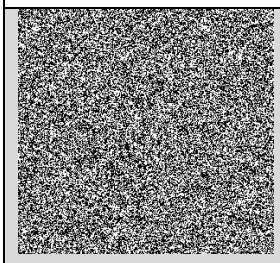
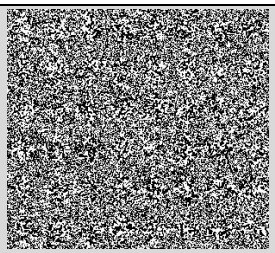
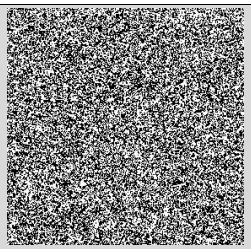
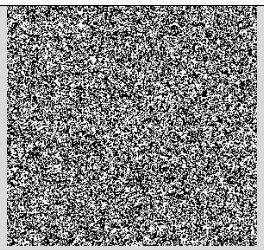
Table 4. StegExpso results

No.	Image name	No. of detected images	Robustness (%)
1	Pepper	0	100%
2	Splash	0	100%
3	Airplane	0	100%
4	Friend	1	88%
5	House	0	100%

All images which were subjected to the stego tool were of size 512×512 pixels. In the test, only one stego image was detected. This was in case where the cover bytes were more than 40% of the image size. It was also found that detecting suspicious images depended on the structure of LSBs of the cover images and statistical characteristics, PoVs of LSBs and histograms which constitute the general pattern for the images. This is attributed to the fact that attackers compare the changes of general pattern with the tables of PoVs of the LSBs of the cover image. If the image characteristics are out of general pattern, they mark image as suspicious image. The hiding mechanism in the proposed algorithm varies while embedding one bit or two bits. As a result, the structure of PoVs of the LSBs changes relative to the embedding case.

Steganalysis performs visual attacks to search for signs of data hiding in the LSB plane. This is done by searching for any difference between cover and stego images in LSB's plane. Random forwardand-backward selection of cover bytes avoids causing any inconsistency in LSB's plane. Table 5 shows the first and second LSB planes for Pepper cover images and their stego images for the case of hiding data at 30% of the cover image size.

Table 5. Bit plane for the two LSBs

Cove Image: Pepper	
Bit 0 Plane	Bit 1 Plane
	
Stego Image at 30%: Pepper	
Bit 0 Plane	Bit 1 Plane
	

7. Comparison with other Algorithms

The proposed method was compared with one-bit LSB substitution and two LSBs substitution techniques. Comparison criteria were PSNR, hiding capacity and bit rate. The results The result in Table 6 show that the quality of the proposed algorithm is greater than the LSB substitution. Table 6 shows the results of the proposed forward-and-backward algorithm in case of hiding secret bits with rates 1.0 and 1.5 bpp. Results for the case of 2.0 and 2.8 bpp are shown in Table 7.

As shown in the tables, the average PSNR from the proposed algorithm for embedding percentage (EP) 12.5%, 18.75%, 25% and 35% were 54.68 dp, 52.05 dp, 50.43 dp and 49.22 dp respectively. Similarly, the embedding capacities were 16,384, 32,768, 49,152, 65,536 and 91,750 bytes for EP equal 12.5%, 18.75%, 25% and 35%, respectively.

The average PSNR of the LSB substitution algorithm for embedding percents equal 12.5% and 25% were 47.29 dp and 43.46 dp respectively (listed in Table 8). Similarly, the embedding capacities of the LSB substitution algorithm were 32,768 and 65,536 bytes for EP equals 12.5% and 25% respectively. These results imply that the average of PSNR of the proposed forward-and-backward algorithm is better than the LSB substitution

algorithm by 5.39 for embedding percentage 12.5% and by 6.97 for embedding percentage 12.5%.

Table 6. Results of Proposed algorithm in 12.5% and 18.5 % of EP

Image	Capacity	BPP	PSNR	Capacity	BPP	PSNR
Pepper	32,768	1.0	55.24	**49,152**	1.5	52.36
Baboon	32,768	1.0	54.99	**49,152**	1.5	52.23
Boat	32,768	1.0	54.87	**49,152**	1.5	52.16
House	32,768	1.0	53.60	**49,152**	1.5	51.45
Average	**32,768**	**1.0**	**54.68**	**49,152**	**1.5**	**52.05**

Table 7. Results of Proposed algorithm in 25% and 35% of EP

Image	Capacity	BPP	PSNR	Capacity	BPP	PSNR
Pepper	65,536	2.0	50.63	91,750	2.8	49.53
Baboon	65,536	2.0	50.56	91,750	2.8	49.42
Boat	65,536	2.0	50.48	91,750	2.8	49.34
House	65,536	2.0	50.05	91,750	2.8	48.6
Average	**65,536**	**2.0**	**50.43**	**91,750**	**2.8**	**49.22**

Table 8. Results of one bit and two-bit LSB substitution [16]

Image	Capacity	BPP	PSNR	Capacity	BPP	PSNR
Pepper	32,768	1.0	46.34	65,536	2.0	44.39
Baboon	32,768	1.0	47.33	65,536	2.0	42.76
Boat	32,768	1.0	47.78	65,536	2.0	43.20
House	32,768	1.0	47.71	65,536	2.0	43.47
Average	**32,768**	**1.0**	**47.29**	**65,536**	**2.0**	**43.46**

8. Conclusion

We have developed a forward-and-backward algorithm for steganography based on an indicator to randomly select cover bytes. Random selection of cover bytes increased the degree of algorithm complexity, detectability and robustness against visual attacks. An additional indicator is used for determining the number of bits to be hidden. The use of the number of secret bits determining indicator increased the capacity of cover bytes. The results of the performed experiments show the quality of the algorithm. Compared with other algorithms, our results were better in terms of image quality. The average PSNR was 54.09 which is a good indicator on the quality of the stego images.

9. References

[1] Morkel, T., Eloff, J. H., & Olivier, M. S. (2005). An overview of image steganography. Paper presented at the ISSA.

[2] Dunbar, B. (2002). A detailed look at Steganographic Techniques and their use in an Open-Systems Environment. Sans Institute, 1.

[3] Lubacz, J., Mazurczyk, W., & Szczypiorski, K. (2014). Principles and overview of network steganography. IEEE Communications Magazine, 52(5), 225-229.

[4] Yadav, P., Mishra, N., & Sharma, S. (2013). A secure video steganography with encryption based on LSB technique. Paper presented at the Computational Intelligence and Computing Research (ICCIC).

[5] Barhoom, T. S., & Mousa, S. M. A. (2015). A steganography LSB technique for hiding image within image using Blowfish encryption algorithm. Int. J. Res. Eng. Sci, 3(3), 61-66.

[6] Provos, N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. IEEE security & privacy, 99(3), 32-44.

[7] Hamid, N., Yahya, A., Ahmad, R. B., & AlQershi, O. M. (2012). Image steganography techniques: an overview. International Journal of Computer Science and Security (IJCSS), 6(3), 168-187.

[8] Karim, S. M., Rahman, M. S., & Hossain, M.I. (2011, December). A new approach for LSB based image steganography using secret key. In 14th International Conference on Computer and

Information Technology (ICCIT 2011) (pp. 286-291). IEEE.

[9] Ren-Er, Y., Zhiwei, Z., Shun, T., & Shilei, D. (2014, January). Image steganography combined with DES encryption preprocessing. In 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation (pp. 323-326). IEEE.

[10] Gutub, A., Ankeer, M., Abu-Ghalioun, M., Shaheen, A., & Alvi, A. (2008). Pixel indicator high capacity technique for RGB image-based Steganography.

[11] Saqer, W., & Barhoom, T. (2016). Steganography and hiding data with indicators-based LSB using a secret key. Engineering, Technology & Applied Science Research, 6(3), 1013-1017.

[12] Akhtar, N., Johri, P., & Khan, S. (2013, September). Enhancing the security and quality of LSB based image steganography. In 2013 5th International Conference and Computational Intelligence and Communication Networks (pp. 385-390). IEEE.

[13] Islam, M. R., Siddiqa, A., Uddin, M. P., Mandal, A. K., & Hossain, M. D. (2014, May). An efficient filtering-based approach improving LSB image steganography using status bit along with AES cryptography. In 2014 International Conference on Informatics, Electronics & Vision (ICIEV) (pp. 1-6). IEEE.

[14] Laha, S., & Roy, R. (2015, December). An improved image steganography scheme with high visual image quality. In 2015 International Conference on Computing, Communication and Security (ICCCS) (pp. 1-6). IEEE.

[15] Rajput, G. G., & Chavan, R. (2017, May). A Novel Approach for Image Steganography based on LSB Technique. In Proceedings of the International Conference on Compute and Data Analysis(pp. 167-170). ACM.

[16] Sahu, A. K., Swain, G., & Babu, E. S. (2018). Digital image steganography using bit flipping. Cybernetics and Information Technologies, 18(1), 69-80.

[17] USC-SIPI. "The USC-SIPI image database.". The University of Southern California. Retrieved on: 25/12/2018, From: http://sipi.usc.edu/database/database.php.

[18] Laskar, S. A., & Hemachandran, K. (2014). A Review on Image Steganalysis techniques for attacking Steganography. International Journal of Engineering Research and Technology, 3(1).