CHAPTER 5

EVALUATION AND COMPARISON RESULTS

5.1 Introduction

This chapter presents the results of the evaluations on the important parameters for CAS algorithm and its components. Next, the comparison results of CAS algorithm against existing works based on each level which are the LSB embedding level, the cover selection level and the audio steganography model level are presented. The results and discussion can be navigated using the illustration in Figure 5.1.

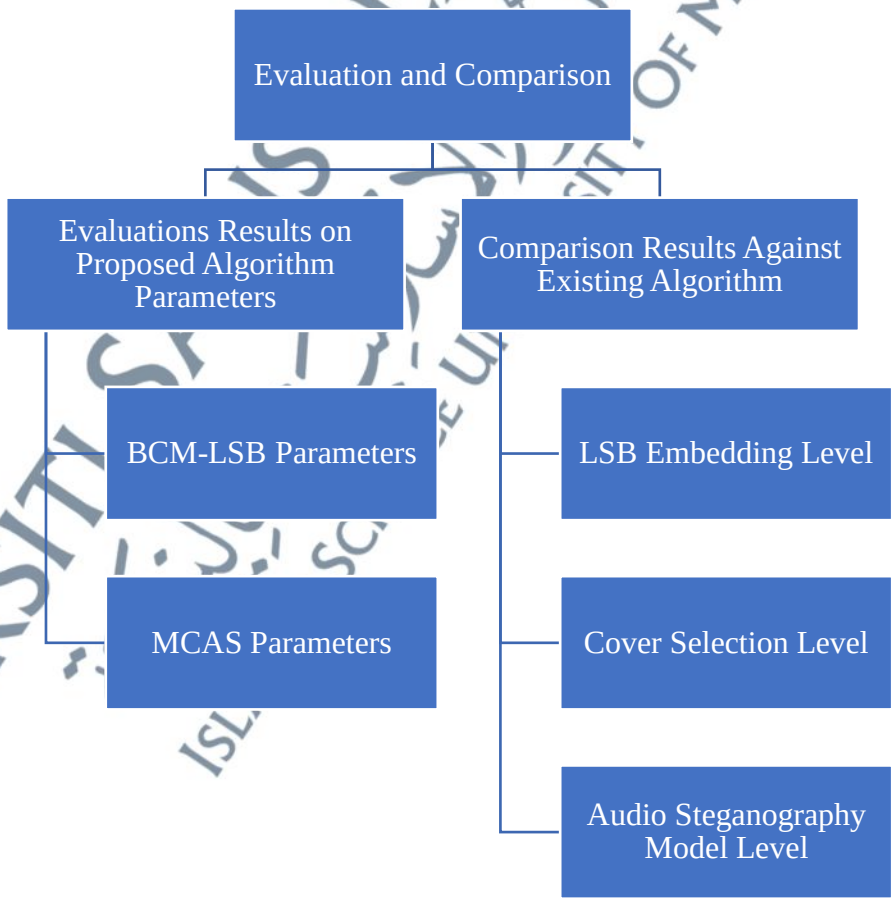


Figure 5.1: Evaluation and Comparison Map

5.2 Evaluations Results on Proposed Algorithm Parameters

This section is divided into two subsections. The first subsection focuses on the evaluations of BCM-LSB parameters' impacts on imperceptibility, capacity, robustness, and dynamic security while the second subsection focuses on the evaluations of MCAS parameters' impact on the performance of stego-file from cover and *bps* selected.

5.2.1 BCM-LSB Parameters Evaluation Results

The following content is organized into four distinct sections, with each one discussing a unique characteristic. Subsection 5.2.1.1 delves into the impact of *bps* on the imperceptibility characteristic, while Subsection 5.2.1.2 analyses the impact of *bps* on capacity. Additionally, Subsection 5.2.1.3 explores how *bps* affects the robustness characteristic, and Subsection 5.2.1.4 examines the impact of key *x*, *n*, and *r* on dynamic security.

5.2.1.1 Impact of *bps* on Imperceptibility Characteristic of BCM-LSB

The objective of this evaluation was to study the BCM-LSB imperceptibility characteristic performance based on *bps* used and select the best *bps* to be used for the comparison experiment. The imperceptibility performances of BCM-LSB variation of embedded *bps* were compared based on SNR, MSE, and PSNR. During the evaluation process, SNR is the primary measure, while MSE and PSNR serve to complement and validate the SNR results. It can also be used to detect any abnormalities in the SNR assessment's output, as the resulting pattern is consistent. To capture the worst imperceptibility performance, maximum capacities of each variation across all the cover audios were applied. Figure 5.2 illustrates the SNR results from the embedding processes performed.

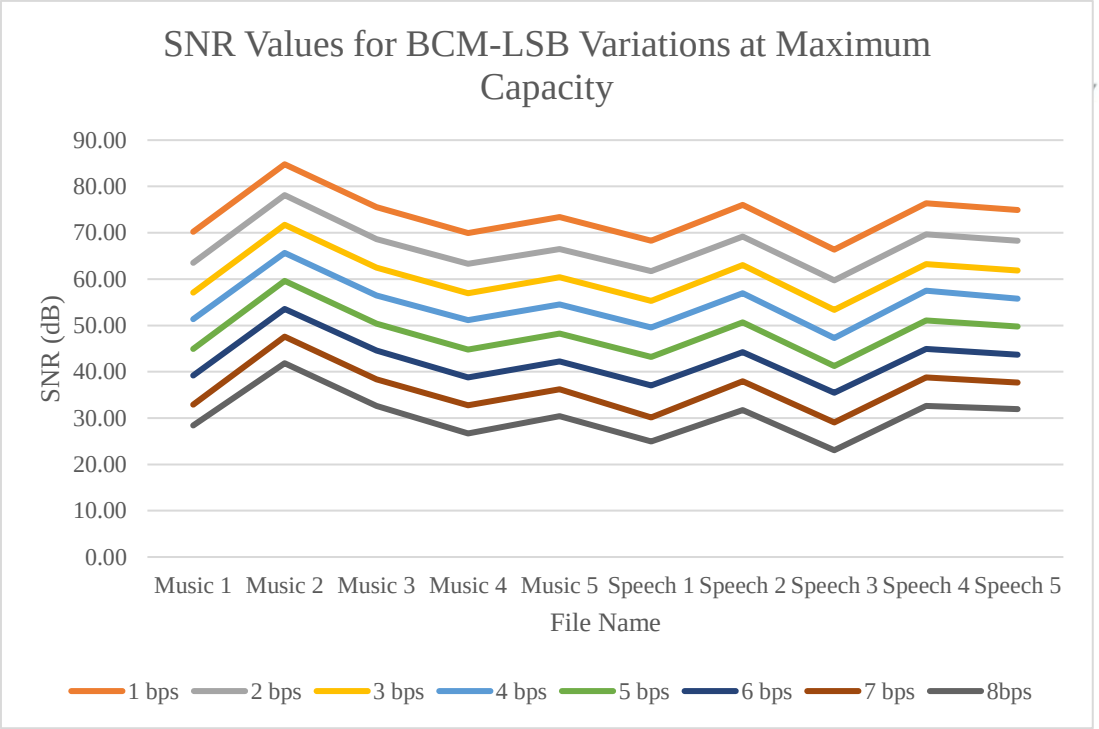


Figure 5.2: SNR Values for Different *bps* with Maximum Capacity per Cover Audio

Figure 5.2 shows the significance of *bps* in terms of imperceptibility performance using the SNR metric. BCM-LSB with 1 *bps* variation was able to achieve the SNR across all the cover audio files used. On the other hand, BCM-LSB with 8 *bps* variation scored the lowest as it got the lowest SNR across all the cover audio files. Higher *bps* indicates a higher possibility of modification to occur on each audio sample. The difference in SNR values across the audio resulted from embedding secret messages with the same variation caused by the different number of audio sample amplitude and the total number of errors from the embedding process. Figure 5.3 presents the MSE values of all BCM-LSB variations for the same experiment.

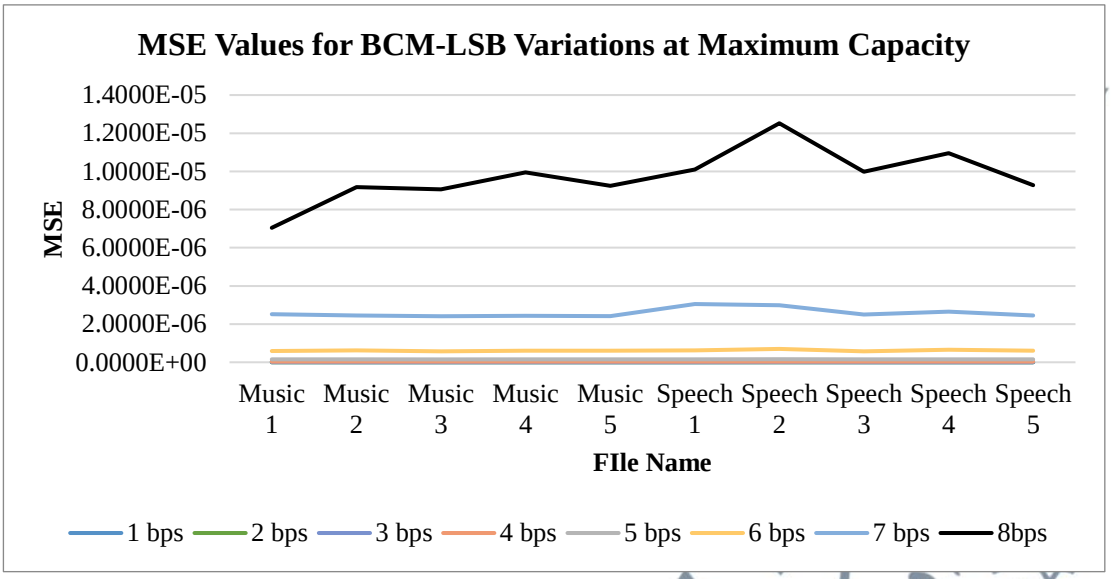


Figure 5.3: MSE Values for Different *bps* with Maximum Capacity per Cover Audio

Based on Figure 5.3, BCM-LSB 8 *bps* variation produced the highest MSE values, followed by BCM-LSB 7 *bps* variation while the lowest MSE values were produced by BCM-LSB 1 *bps*. Figure 5.4 presents the PSNR values of all BCM-LSB variations for the same experiment.

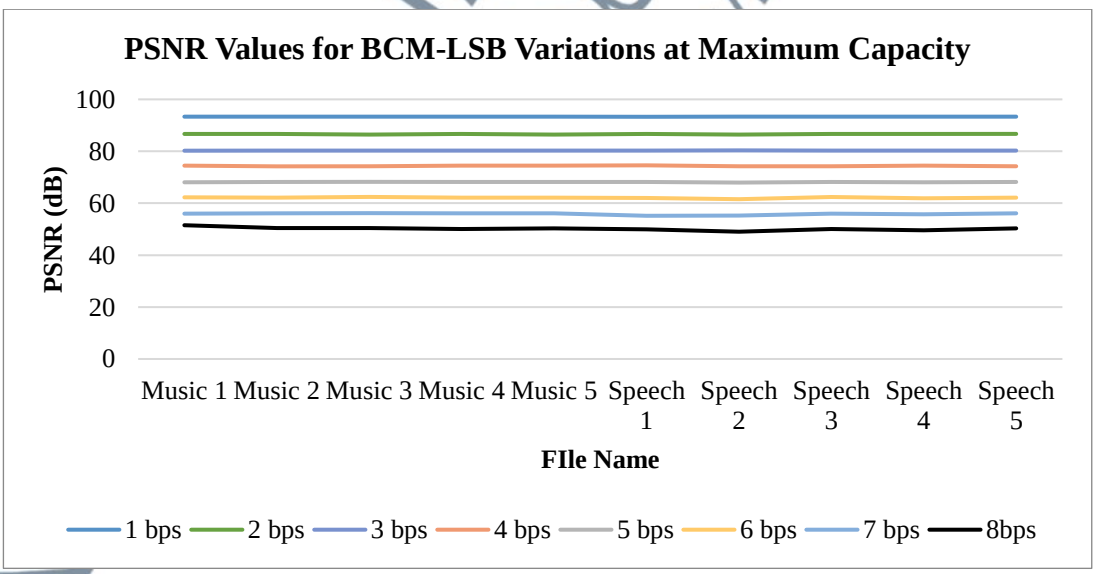


Figure 5.4: PSNR Values for Different *bps* with Maximum Capacity per Cover Audio

As seen in Figure 5.4, PSNR exhibited the same patterns as SNR. However, PSNR rates were more tolerant to noise than SNR rates. The values are almost similar across

different audio files used when embedding the same *bps* compared to the SNR values. The values are similar because the formulation used the highest sample values for the PSNR calculation compared to using current sample values in SNR. The minimum PSNR value for the worst case of BCM-LSB was approximately 50 dB while the maximum PSNR value was approximately 93 dB. It is critical to note that the SNR, MSE, and PSNR all produced identical significant results overall, which shows that these evaluations are consistent and produced the same conclusions.

In summary, all variations are able to achieve an acceptable level of imperceptibility even at maximum capacity as the minimum value for PSNR is above 30 dB while the minimum value for SNR is above 20 dB (P. Li et al., 2014). The highest imperceptibility level is related to the highest SNR and PSNR as well as to the lowest MSE. The BCM-LSB with a one (1) *bps* variation has the highest imperceptibility because of the lowest modification during the embedding process (Indrayani, 2020) compared to the BCM-LSB, which has an eight (8) *bps* variation with the lowest imperceptibility characteristic. Hence, BCM-LSB with 1 *bps* is used for imperceptibility comparison against other existing LSB embedding algorithms.

5.2.1.2 Impact of *bps* on Capacity Characteristic of BCM-LSB

The objective of this evaluation was to study the BCM-LSB capacity characteristic performance based on *bps* used and select the best *bps* to be used for algorithm comparison experiment. These BCM-LSB variations of embedded *bps* were compared in terms of maximum size in kilobyte (KB) of the secret message which managed to be embedded in the cover audio file. The maximum capacity results are presented in Table 5.1.

Table 5.1: Maximum Capacity Values for Different *bps* per Cover Audio.

File Name	Maximum Capacity (KB)							
	1 *bps*	2 *bps*	3 *bps*	4 *bps*	5 *bps*	6 *bps*	7 *bps*	*8bps*
Music 1	5.507	11.014	16.521	22.028	27.535	33.042	38.549	44.056
Music 2	11.8130	23.626	35.439	47.252	59.065	70.878	82.691	94.504
Music 3	17.38	34.76	52.14	69.52	86.9	104.28	121.66	139.04
Music 4	29.3360	58.672	88.008	117.344	146.68	176.016	205.352	234.688
Music 5	49.607	99.214	148.821	198.428	248.035	297.642	347.249	396.856
Speech 1	5.507	11.014	16.521	22.028	27.535	33.042	38.549	44.056
Speech 2	11.387	22.774	34.161	45.548	56.935	68.322	79.709	91.096
Speech 3	17.579	35.158	52.737	70.316	87.895	105.474	123.053	140.632
Speech 4	30.235	60.47	90.705	120.94	151.175	181.41	211.645	241.88
Speech 5	51.625	103.25	154.875	206.5	258.125	309.75	361.375	413
Min	5.507	11.014	16.521	22.028	27.535	33.042	38.549	44.056
Max	51.625	103.25	154.875	206.5	258.125	309.75	361.375	413
Average	22.9976	45.9952	68.9928	91.9904	114.988	137.9856	160.9832	183.9808

Table 5.1 shows that the BCM-LSB 8 *bps* variation achieved the highest maximum capacity followed by BCM-LSB 7 *bps*, BCM-LSB 6 *bps* BCM-LSB 5 *bps* and the trend went down until BCM-LSB 1 *bps*. This is because more bits of secret message are embedded with higher bps used hence the capacity performance is increased (Cvejic & Seppanen, 2002; Indrayani, 2020). In addition, based on this table, the maximum capacity also demonstrated a linear relationship with the length of the cover audio. The longer the cover audio used, the higher the capacity of the BCM-LSB.

In summary, BCM-LSB with 8 *bps* produces high-capacity performance. Hence, 8 *bps* is used for capacity comparison against other existing LSB embedding algorithms.

5.2.1.3 Impact of *bps* on Robustness Characteristic of BCM-LSB

The objective of this evaluation was to study the BCM-LSB robustness characteristic performance based on *bps* used and select the best *bps* to be used for algorithm comparison experiment. This section compares the robustness of BCM-LSB with different values of *bps* parameter. The experiment demonstrates the robustness of the BCM-LSB against various loads of AWGN. In this experiment, half of each BCM-LSB variations algorithm's maximal capacity was embedded. The recovery procedure is then executed by utilising the recovery portion of each technique to locate the embedded message. The performance of the offered techniques is captured using the three prior cover audios. Figure 5.5, Figure 5.6 and Figure 5.7 depict the effects of the AWGN attack on the first, second and third cover audio file.

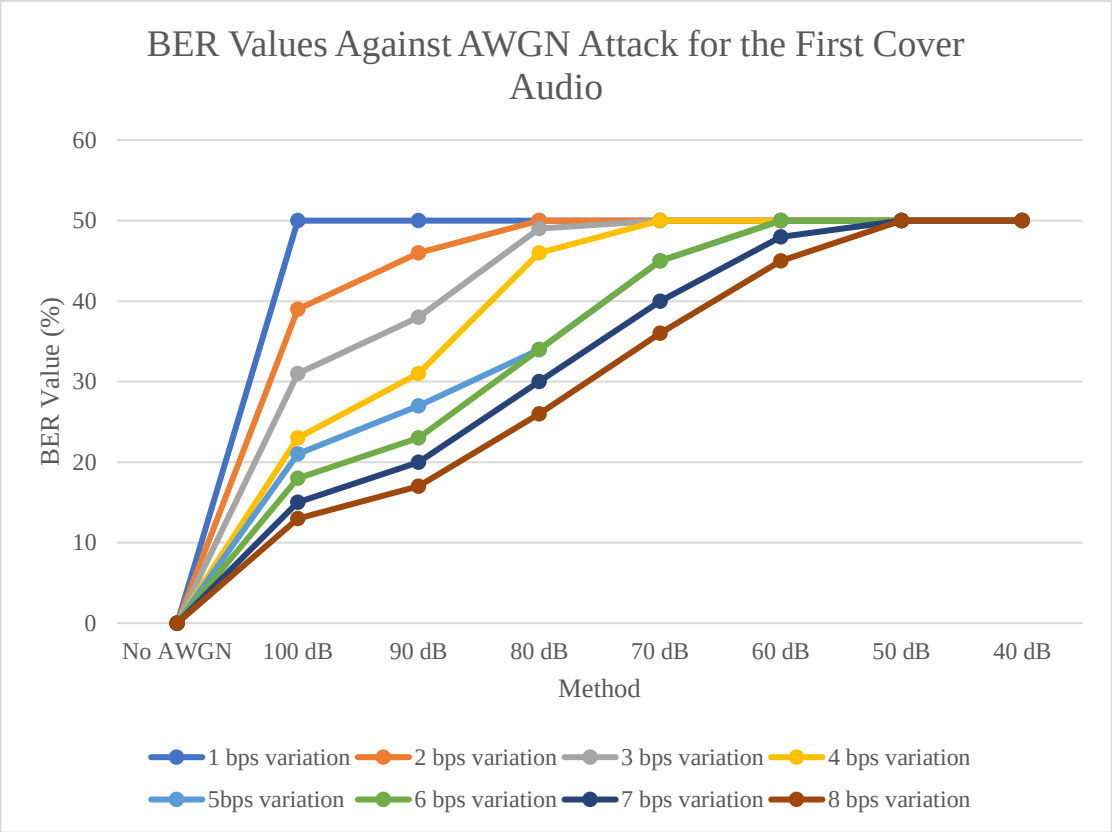


Figure 5.5: BER Values Against AWGN Attack for First Cover Audio

Figure 5.5 highlights the lowest BER value produced by the 8 *bps* BCM-LSB variation while the highest BER value produced by the 1 *bps* BCM-LSB variations. 1 *bps* BCM-LSB already maxed the BER value (50%) at weakest AWGN attack setting which was set for 100 dB. At similar AWGN attack setting, 8 *bps* BCM-LSB variation achieved the lowest BER value at around 13%. At 80dB AWGN attack setting, stego-file produced by BCM-LSB with 1 *bps*, 2 *bps*, 3 *bps* and 4 *bps* already reached loss point as BER already exceed 45% (Djebbar et al., 2010). All *bps* variations not able to withstand 60dB, 50dB and 40dB AWGN attack settings. The highest settings that the BCM-LSB algorithm managed to withstand is at 70dB with using 7 *bps* and 8 *bps*. Generally BCM-LSB with *8 bps* has the highest robustness in all AWGN attack setting compared to other *bps* variations as embedding at higher level could improve the robustness characteristic performance (Hosny et al., 2019).

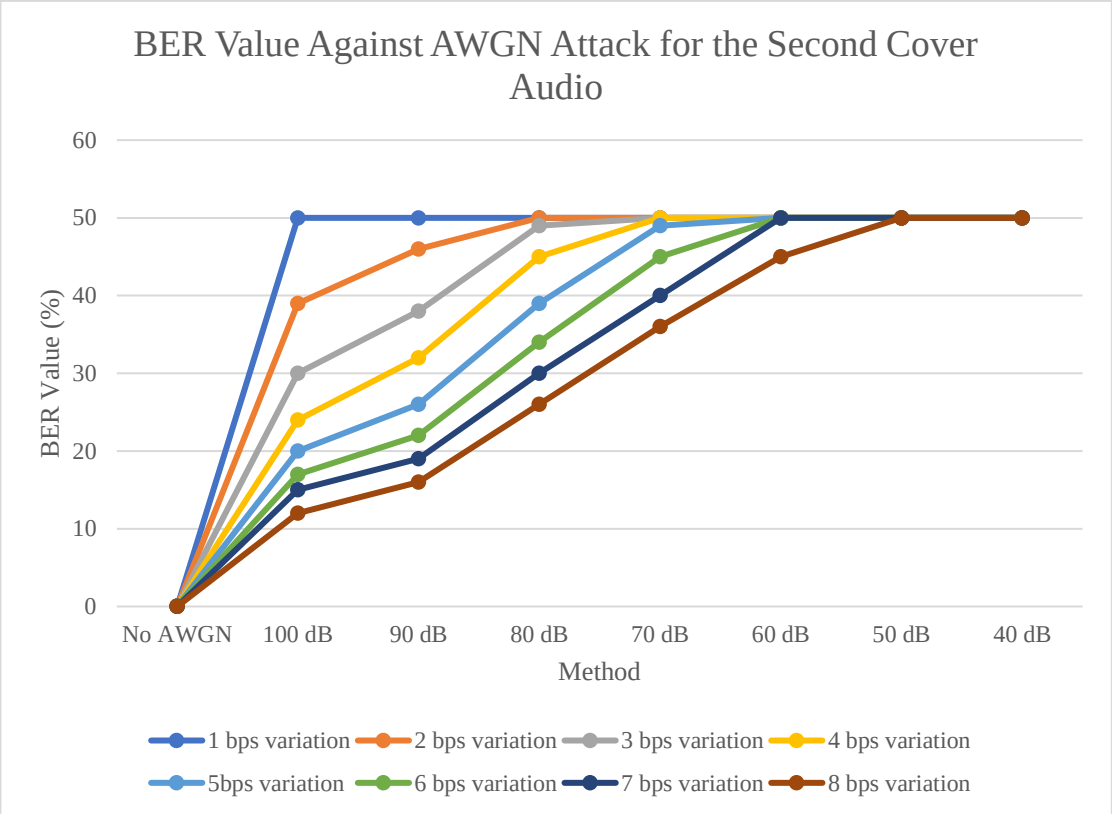


Figure 5.6: BER Values Against AWGN Attack for Second Cover Audio

Figure 5.6 shows similar pattern as in Figure 5.5. BCM-LSB with 8 *bps* variation has the lowest BER compared to other *bps* variations in all attack except at 50dB and 40dB. Although at 60dB AWGN attack setting, 8 *bps* BCM-LSB variation records 45%, which already reached the loss point. Therefore, the stego-file produced cannot be used as it is destroyed in term of the bit stored due the secret message that flipped as much as 45%. Therefore, all *bps* variations were not able to withstand 60dB, 50dB and 40dB AWGN attack settings. The highest settings that the BCM-LSB algorithm managed to withstand is at 70dB with using 7 *bps* and 8 *bps*. Similar conclusion can be drawn from this result which is BCM-LSB with *8 bps* has the highest robustness in all AWGN attack setting compared to other *bps* variations as embedding at a higher level could improve the robustness characteristic performance (Hosny et al., 2019).

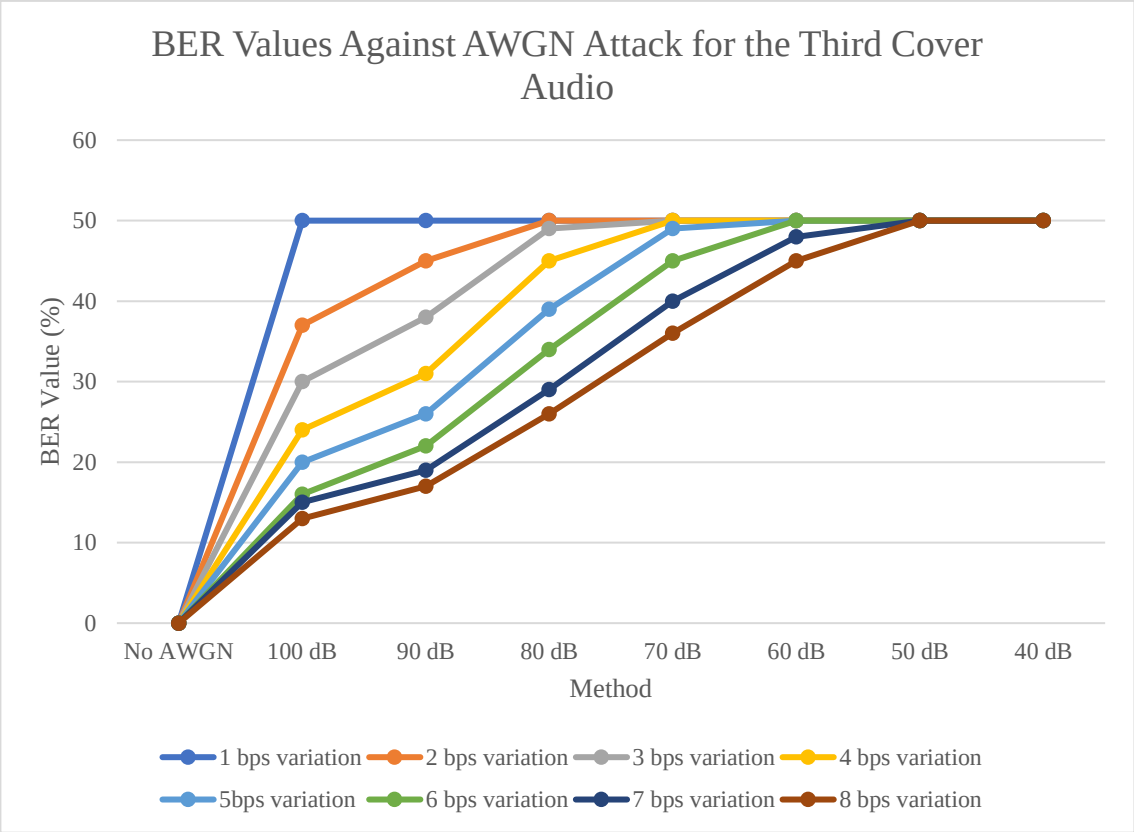


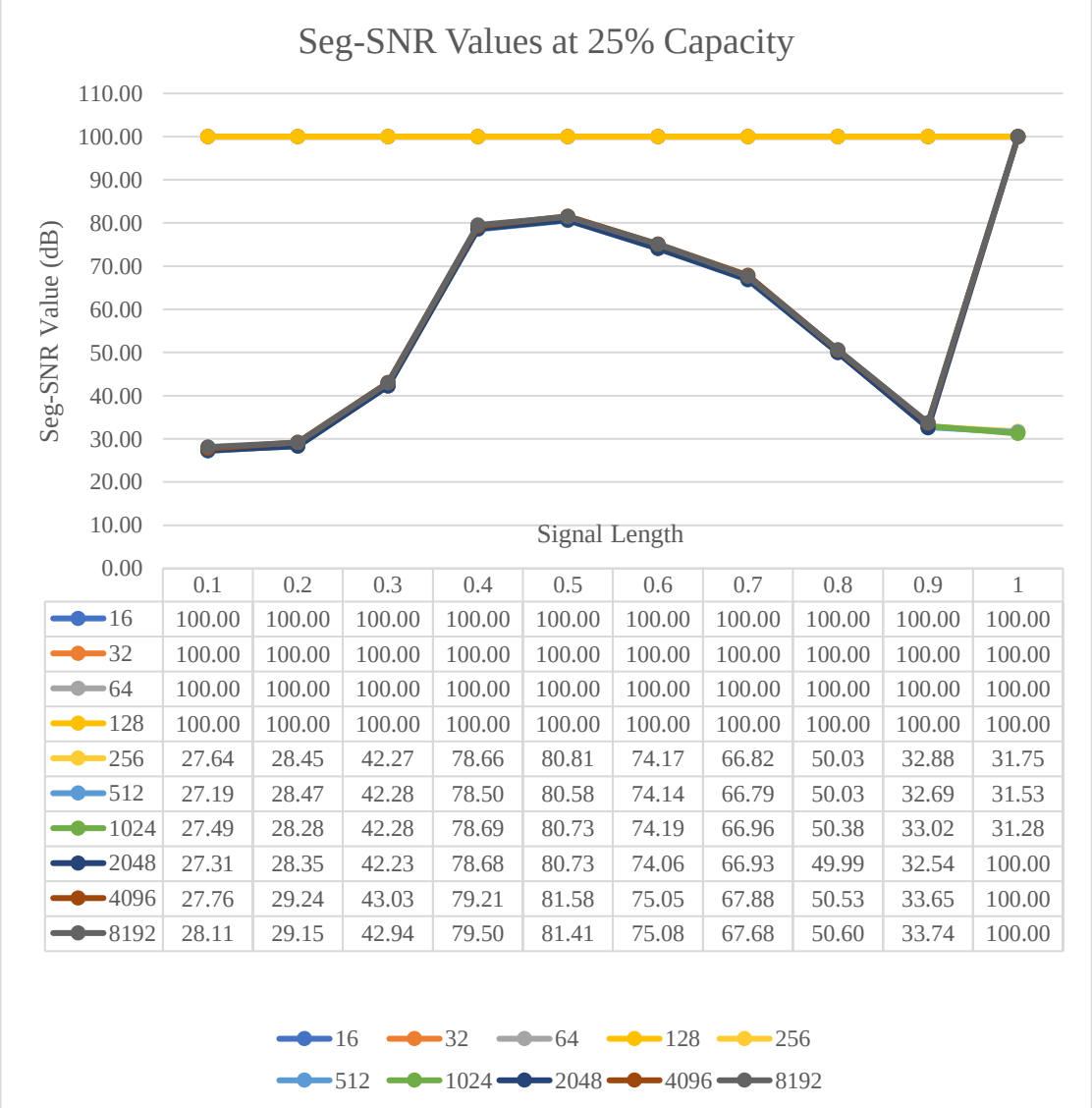
Figure 5.7: BER Values Against AWGN Attack for Third Cover Audio

Again, Figure 5.7 shows similar pattern as in Figure 5.5 and Figure 5.6. Although each figure has a slightly different value, this is due to the differing number of flipped audio sample binary values during the embedding process, resulting in different outputs with similar AWGN attack settings. In summary, BCM-LSB 8 *bps* achieves the highest robustness compared to other *bps* used which in line with state-of-art knowledge on increasing the robustness characteristic – embed at higher level of LSB.

5.2.1.4 Impact of *bps* on Dynamic Security Characteristic of BCM-LSB

As described earlier in Chapter 4, the BCM-LSB has a set of keys used for the creation of the chaotic block. To evaluate the performance of the dynamic security between the set of keys, two separate Seg-SNR Spike Tests were conducted. The first test was conducted against a different number of keys *n*, which represent the number of

blocks used, while the second test was conducted against different pairs of keys that determine the location of the initial point of embedding in each block which are *key x* and *key r*. The result of the first test is presented in Figure 5.8 while the result of the second test is presented in Figure 5.9. The seg-SNR values were substituted with hundred (100) dB to denote the embedding process did not occur at the segment.

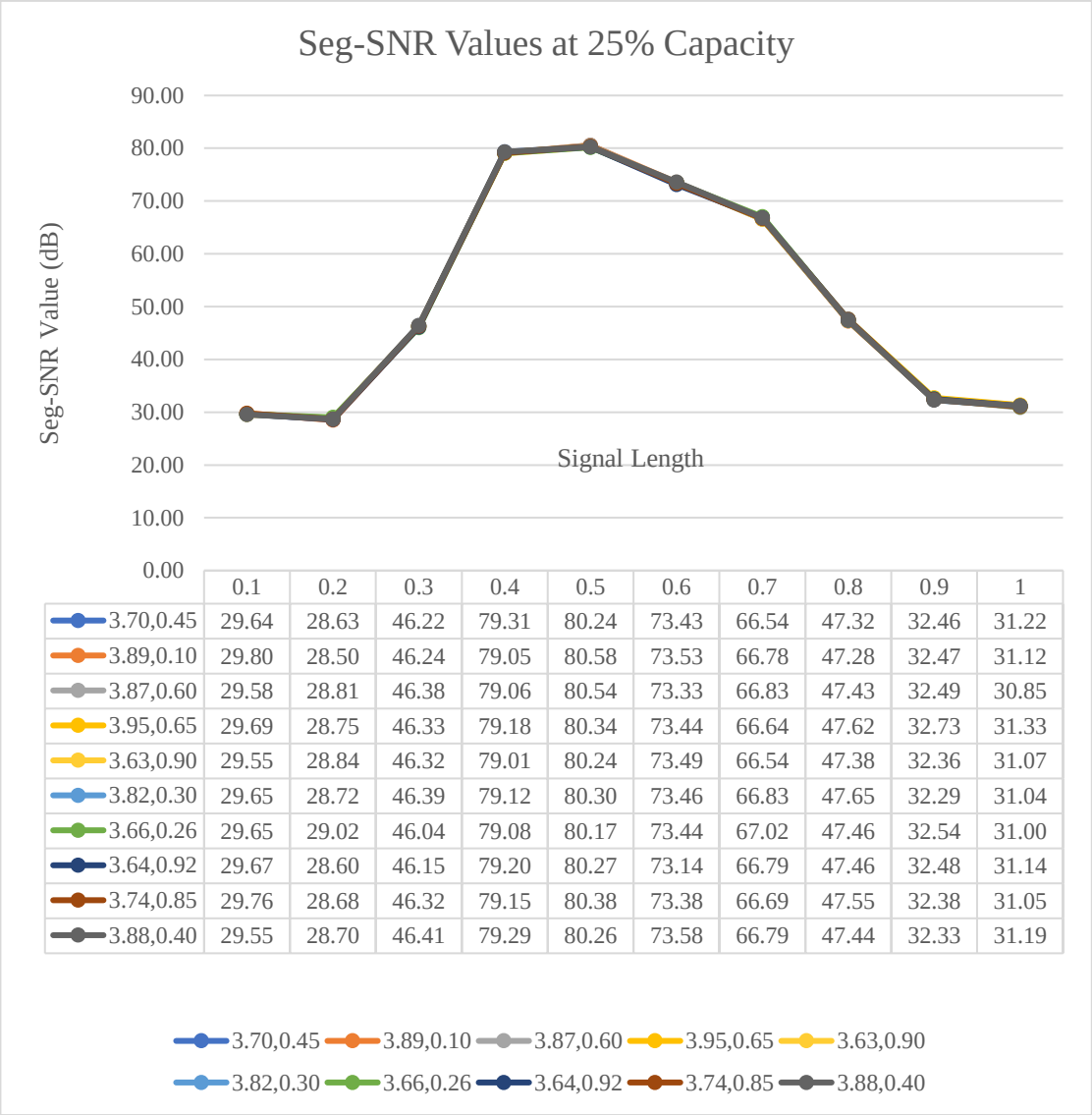


	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
16	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
32	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
64	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
128	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
256	27.64	28.45	42.27	78.66	80.81	74.17	66.82	50.03	32.88	31.75
512	27.19	28.47	42.28	78.50	80.58	74.14	66.79	50.03	32.69	31.53
1024	27.49	28.28	42.28	78.69	80.73	74.19	66.96	50.38	33.02	31.28
2048	27.31	28.35	42.23	78.68	80.73	74.06	66.93	49.99	32.54	100.00
4096	27.76	29.24	43.03	79.21	81.58	75.05	67.88	50.53	33.65	100.00
8192	28.11	29.15	42.94	79.50	81.41	75.08	67.68	50.60	33.74	100.00

Figure 5.8: Seg-SNR Values at 25% Capacity

In the first test, the *key x* and *key r* were set at 0.65 and 3.95, respectively. The *bps* was set to one (1) and the 1-second audio was used. The seg-SNR value, which has

an infinity value, was replaced by 100 dB for the sake of visual representation in Figure 5.8. Based on Figure 5.8, a mixed pattern was produced from a different number of blocks used. The key values of n = 16, 32, 64, and 128 were unusable due to infinite values throughout all segments, resulting in the worst-case scenario. Next, *key n* = 256, 512 and 1024 obtained the best possible scenario using the current cover audio file. Although there was a sudden spike between segment 3 and segment 4, it was still acceptable compared to the sudden spike to infinity. This spike shows that there is a huge difference of amplitudes between segments which affect the SNR values calculated. Lastly, *key n* = 2048, 4096 and 8192 exhibited spikes in the last segment. In conclusion, a precise block number of either 256, 512 or 1024 is suitable to be used as key *n* to embed the secret message over the cover audio.



	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
3.70,0.45	29.64	28.63	46.22	79.31	80.24	73.43	66.54	47.32	32.46	31.22
3.89,0.10	29.80	28.50	46.24	79.05	80.58	73.53	66.78	47.28	32.47	31.12
3.87,0.60	29.58	28.81	46.38	79.06	80.54	73.33	66.83	47.43	32.49	30.85
3.95,0.65	29.69	28.75	46.33	79.18	80.34	73.44	66.64	47.62	32.73	31.33
3.63,0.90	29.55	28.84	46.32	79.01	80.24	73.49	66.54	47.38	32.36	31.07
3.82,0.30	29.65	28.72	46.39	79.12	80.30	73.46	66.83	47.65	32.29	31.04
3.66,0.26	29.65	29.02	46.04	79.08	80.17	73.44	67.02	47.46	32.54	31.00
3.64,0.92	29.67	28.60	46.15	79.20	80.27	73.14	66.79	47.46	32.48	31.14
3.74,0.85	29.76	28.68	46.32	79.15	80.38	73.38	66.69	47.55	32.38	31.05
3.88,0.40	29.55	28.70	46.41	79.29	80.26	73.58	66.79	47.44	32.33	31.19

Figure 5.9: Seg-SNR Values at 25% Capacity

In the second test, the *key n* was set to 1024. The *bps* was set to one (1) and the 1-second audio duration was used. The seg-SNR value which has an infinity value was replaced by 100 dB for the sake of visual representation in Figure 5.9. Based on Figure 5.9, there was no difference between the usage of *key x* and *key r*. All key pairs managed to distribute the secret message throughout the cover audio without any value that spiked up to infinity. It is important to carefully choose the pair of keys even though they do not affect dynamic security. This is because the keys still need to maintain randomization elements that benefit the steganographic algorithm. By doing so, the

audio steganography becomes more robust and is protected from direct retrieval attacks (Ahmed A Alsabhany, 2019; Kanhe et al., 2015). In summary, any value of key *x* and *r* can be used as a parameter as long as within the range stated in Chapter 3 and Chapter 4. On the other hand, only certain value of key *n* can be used which are 256, 512 or 1024. Based on these experiments, key *x*, *r* and *n* with values of 0.92, 3.64 and 1024 respectively are chosen for evaluation conducted for MCAS and comparison against existing methods at LSB embedding layer, cover selection layer and audio steganography model layer. They are chosen to bring the best of dynamic security performance.

5.2.2 MCAS Parameters Evaluation Results

This section is divided into three parts which discussed evaluations on three parameters settings. Subsection 5.2.2.1 discusses the impact of number of generations used *maxIt* on performance of stego-file produced while Subsection 5.2.2.2 discusses the impact of number of populations, *nPop* used on performance of stego-file and lastly, Subsection 5.2.2.3 discusses the impact of the usage of remove duplicated solution in population function on stego-file performance.

5.2.2.1 Impact of Number of Generation Used

MCAS with different variations generation number used were compared in terms of the quantity and the quality of the solution found. The objective of this evaluation was to compare the performance of stego-files produced based on parameter, number of generations used, *maxIt* and select the best *maxIt* to be used for algorithm comparison experiment. This experiment is conducted by collecting the solutions produced by the MCAS using the number of 10, 15 and 20 generations. Next, the solutions produced are

collected and compared to each other. The results with the usage of number of generations = 10, number of generations = 15 and number of generations = 20 are shown in Table 5.2. Table 5.3, Table 5.4. Table 5.5 summarizes the result of the comparison between solutions produced from different number of generations.

Table 5.2: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Generation = 10

Round	**NPS**	**NDS**	$\boldsymbol{Best_{Sol}}$ **(%)**	$\boldsymbol{Worst_{Sol}}$**(%)**
Round 1	21	0	77.895	38.918
Round 2	21	0	77.895	38.918
Round 3	21	0	77.895	38.918

Table 5.3: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Generation = 15

Round	***NPS***	**NDS**	$\boldsymbol{Best_{Sol}}$ **(%)**	$\boldsymbol{Worst_{Sol}}$**(%)**
Round 1	21	0	77.895	38.918
Round 2	21	0	77.895	38.918
Round 3	21	0	77.895	38.918

Table 5.4: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Generation = 20

Round	***NPS***	**NDS**	$\boldsymbol{Best_{Sol}}$ **(%)**	$\boldsymbol{Worst_{Sol}}$**(%)**
Round 1	21	0	77.895	38.918
Round 2	21	0	77.895	38.918
Round 3	21	0	77.895	38.918

Table 5.5: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Generation	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
10	21	0	77.895	38.918
15	21	0	77.895	38.918
20	21	0	7.7895	38.918

Based on Table 5.2, Table 5.3, Table 5.4 and Table 5.5, all number of generation variations managed to obtain similar results of *NPS*, $Best_{Sol}$ and $Worst_{Sol}$. In the experiment, it was observed that the MCAS was able to identify the optimal solutions with the least number of generation variations. As a result, there was no significant difference between the parameter setups. Although the output was the same for each value used, this experiment is still important because determining a feasible value for the number of generations is crucial as different multi-objective problems require different values (Campos Ciro et al., 2016; Vachhani, 2016). In summary, 10, 15 and 20 can be used as the value for number of generations, *maxIt*. However, this research picks 20 as number of generation value to ensure all satisfied solutions were found and no premature solutions were produced from limited number of generations used.

5.2.2.2 Impact of Size of Population Used

MCAS with different variations size of population were then compared in terms of the quantity and the quality of the solution found. The objective of this evaluation was to compare the performance of stego-files produced based on parameter, size of population used, *nPop* and select the best *nPop* to be used for algorithm comparison experiment. This experiment is conducted by collecting the solutions produced by the MCAS using size of population of 100, 200 and 300. Next the solutions produced are

collected and compared to each other. The results with the usage of size of population = 100, size of population = 200 and size of population = 300 are shown in Table 5.6, Table 5.7 and Table 5.8.

Table 5.6: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Population = 100

Round	**NPS**	**NDS**	$\boldsymbol{Best_{Sol}}$ (%)	$\boldsymbol{Worst_{Sol}}$(%)
Round 1	14	0	68.856	35.508
Round 2	14	0	68.856	35.508
Round 3	14	0	68.856	35.508

Table 5.7: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Population = 200

Round	***NPS***	**NDS**	$\boldsymbol{Best_{Sol}}$ **(%)**	$\boldsymbol{Worst_{Sol}}$**(%)**
Round 1	14	0	68.856	35.508
Round 2	14	0	68.856	35.508
Round 3	14	0	68.856	35.508

Table 5.8: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Population = 300

Round	***NPS***	**NDS**	$\boldsymbol{Best_{Sol}}$ **(%)**	$\boldsymbol{Worst_{Sol}}$**(%)**
Round 1	21	0	77.895	37.751
Round 2	21	0	77.895	37.751
Round 3	21	0	77.895	37.751

Based on Table 5.6, Table 5.7, and Table 5.8, all sizes of population variation managed to obtain similar results of *NPS*, $Best_{Sol}$ and $Worst_{Sol}$ during each round. However, since not each populations found similar solutions in their Pareto front, a new

population was created by merging all these three Pareto fronts to find new pareto front. Table 5.9 then shows the results of the new population created.

Table 5.9: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Population	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
100	12	2	77.895	66.460
200	12	2	77.895	66.460
300	21	0	77.895	38.918

According to Table 5.9, the population of size 300 achieved the highest NPS while obtaining the lowest NDS. This suggests that the 300-sized population discovered the optimal solution, which cannot be outperformed by any other solutions from different-sized populations. Moreover, it was discovered that all populations, irrespective of their size, generated the same $Best_{Sol}$. However, the 300-sized population had the least $Worst_{Sol}$. This simply means that the solution found from a 300-sized population produced from $Best_{Sol}$ to $Worst_{Sol}$ as two dominated solutions found by 100-sized and 200-sized population were pushed from the new Pareto front. Consequently, it is recommended to use a 300-sized population as a parameter value compared to the other two sizes for improved MCAS during comparison experiment against other existing cover selection algorithm.

5.2.2.3 Removal Duplicated Solutions Algorithm Used

MCAS with and without removal duplicated solutions algorithm is compared in terms of the quantity and the quality of the solution found. The objective of this evaluation was to compare the performance of stego-files produced based on parameter, usage of removal duplicated solution algorithm and select either with or without

removal duplicated solutions algorithm to be used for algorithm comparison experiment. This experiment is conducted by collecting the solutions produced by the MCAS with and without removal duplicated solutions algorithm. Next, the solutions produced are compared to each other. The results with and without removal duplicated solutions algorithm are shown in Table 5.10 and Table 5.11.

Table 5.10: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ With Removal Duplicated Solutions Algorithm

Round	NPS	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	21	0	77.895	37.751
Round 2	21	0	77.895	37.751
Round 3	21	0	77.895	37.751

Table 5.11: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Without Removal Duplicated Solutions Algorithm

Round	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	300	0	77.895	37.751
Round 2	300	0	77.895	37.751
Round 3	300	0	77.895	37.751

Based on Table 5.10 and Table 5.11, MCAS with and without duplicated solutions algorithm managed to obtain similar results of *NDS*, $Best_{Sol}$ and $Worst_{Sol}$ in each round. However, without the removal duplicated solutions algorithm, the number of NPS was 300 for all three rounds. As each set did not find the same solutions in their Pareto front, a new population was created by merging all these two Pareto fronts to find new Pareto front. Table 5.12 shows the results of the new population created.

Table 5.12: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Removal Duplicated Algorithm	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
With	300	0	77.895	37.751
Without	21	0	77.895	37.751

Based on Table 5.12, both with and without the removal duplicated solution algorithm produced the same NDS, $Best_{Sol}$ and $Worst_{Sol}$. Although there was a huge difference in NPS, it did not affect the performance as the solutions found were duplicated solutions. Therefore, the implementation of the removal duplicated algorithm did not impact the solution found. Although the output from MCAS with and without removal duplicated solutions algorithm used are similar, this experiment is still important because finding the impact from both is critical as different multi-objective problem may either requires or does not requires the duplicate solution removal algorithms (Beume et al., 2009; Cremene et al., 2015). In summary, MCAS is good either with or without this algorithm. In order to mitigate the risk of encountering redundant solutions and to guarantee the discovery of all viable options without generating premature ones, this research has opted to utilize MCAS in conjunction with a duplicate-removal algorithm as a precautionary measure.

5.3 Comparison Results Against Existing Algorithm

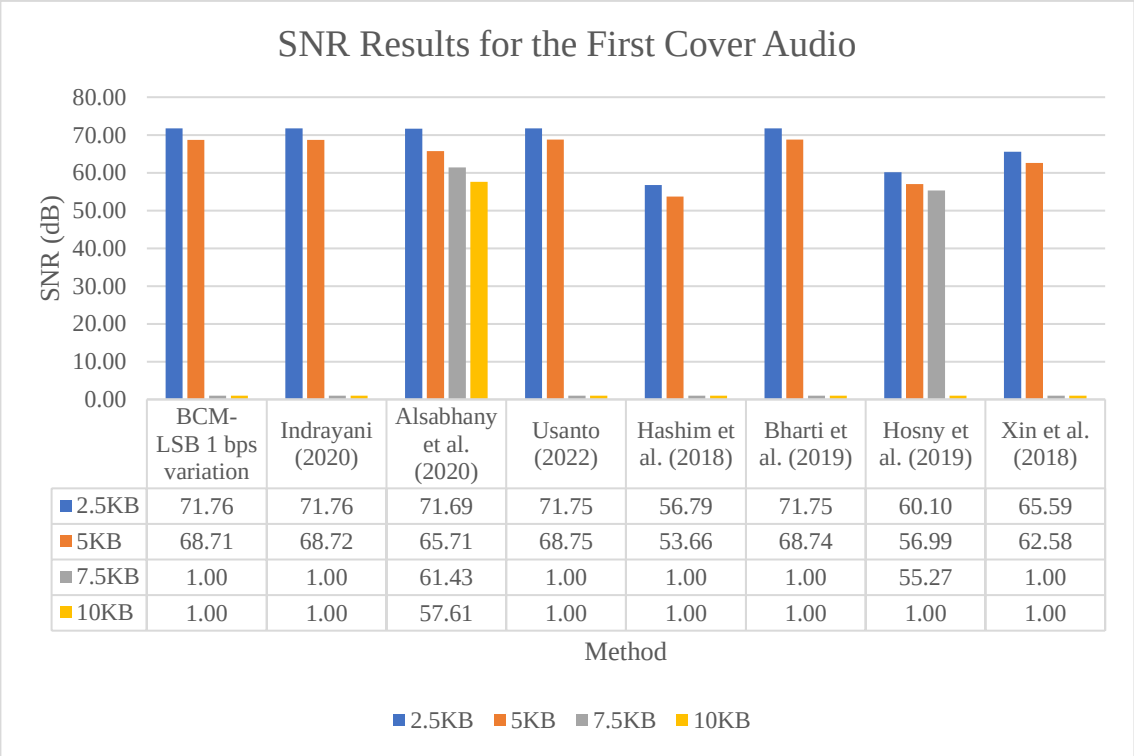
This section is divided into three subsections. The first subsection focuses on the comparison evaluations of proposed algorithm against existing algorithm at LSB embedding level while the second subsection focuses on the comparison evaluations of proposed algorithm against existing algorithm at cover selection level. Lastly, the third subsection focuses on the comparison evaluations of proposed algorithm against existing algorithm at audio steganography model level.

5.3.1 LSB Embedding Level Comparison Results

This section is divided into four parts, each covering one of the four characteristics. In Subsection 5.2.1.1, we discuss the comparison between BCM-LSB and existing LSB embedding algorithms based on their imperceptibility. Subsection 5.2.1.2 focuses on the comparison of BCM-LSB and existing LSB embedding algorithms based on their capacity. Then, in Subsection 5.2.1.3, we compare BCM-LSB and existing LSB embedding algorithms with respect to robustness. Finally, subsection 5.2.1.4 compares BCM-LSB and existing LSB embedding algorithms based on their dynamic security performance.

5.3.1.1 Imperceptibility Characteristic Performance Comparison Results

The objective of this evaluation was to compare the BCM-LSB algorithm's imperceptibility against existing algorithms in terms of SNR, MSE, and PSNR using the first, second, and third cover audio. Figure 5.10 depicts the SNR evaluation for the first cover audio. This cover audio file was selected because it demonstrated the possibilities of the approach in precisely one second. This sample's range of secret message sizes involved 2.5 KB, 5 KB, 7.5 KB, and 10 KB, depending on the capacity constraints of the involved algorithms. Due to capacity limitations, some approaches failed to embed the secret message. An SNR value of 1 in this figure is used to indicate embedding failure due to capacity constraints.



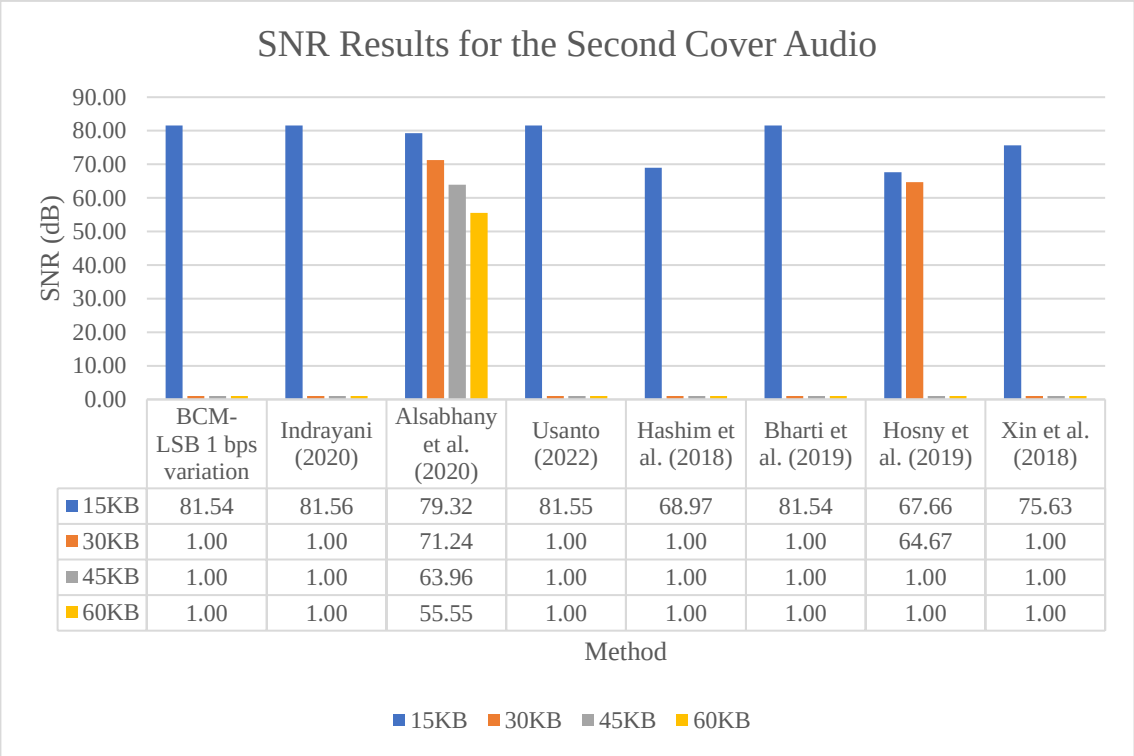
	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
2.5KB	71.76	71.76	71.69	71.75	56.79	71.75	60.10	65.59
5KB	68.71	68.72	65.71	68.75	53.66	68.74	56.99	62.58
7.5KB	1.00	1.00	61.43	1.00	1.00	1.00	55.27	1.00
10KB	1.00	1.00	57.61	1.00	1.00	1.00	1.00	1.00

Figure 5.10: SNR Results for the First Cover Audio

Figure 5.10 shows that several algorithms which are BCM-LSB 1 *bps* variation, Indrayani (2020), Usanto (2022), Hashim et al. (2018), Bharti et al. (2019), Hosny et al. (2019) and Xin et al. (2018) failed to embed some of the secret messages due to capacity limitations. This experiment highlighted the superiority of the BCM-LSB with 1 *bps* variation and the works by Indrayani, (2020) and Bharti et al. (2019) with SNR values of both marked at 71.8 dB when embedded 2.5 KB secret message. It is worth noting that Alsabhany et al. (2020) achieved nearly identical results using the BCM-LSB algorithm with a 1 *bps* embedding rate when embedding secret messages of 2.5 KB and 5 KB, respectively. This is because the algorithm adapts by embedding at a higher *bps* when a larger message size needs to be hidden. In this experiment, this algorithm embedded approximately 1 *bps* at 2.5 KB and 5 KB which is why it produced similar results with the BCM-LSB 1 *bps*. It is also worth noting that other algorithms such as Hashim et al. (2018), Hosny et al. (2019), and Xin et al., (2018) have only been able to

embed 2.5 KB and 5 KB of total secret messages. Although they were only able to embed the secret messages of these two sizes, they produced significantly lower SNR compared to the BCM-LSB 1 *bps*. This is because they embed at a higher level of LSB which reduces the SNR value in general. On the other hand, Usanto (2022) also produced lower SNR values, but it was not statistically significant. This was because it was also embedded at the lowest LSB, similar to BCM-LSB 1 *bps*. The only reason for this algorithm to produce lower values was because of the different way of embedding, which resulted in varying locations of embedding that led to a different number of bits flipping during the embedding process.

Figure 5.11 depicts the SNR evaluation findings for the second audio sample. This sample's range of secret message sizes involves 15 KB, 30 KB, 45 KB, and 60 KB. Depending on the capacity constraints of the involved algorithms. Due to their capacity limitations, a few approaches failed to embed the secret message. An SNR value of 1 in this figure is used to indicate embedding failure due to capacity constraints. During the implementation of BCM-LSB, only 5 KB payloads can be embedded with 1 *bps* variation. However, to assess the complete performance of other existing LSB embedding algorithms that can embed more than 5 KB, 30 KB, 45 KB, and 60 KB payloads were used.



	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
15KB	81.54	81.56	79.32	81.55	68.97	81.54	67.66	75.63
30KB	1.00	1.00	71.24	1.00	1.00	1.00	64.67	1.00
45KB	1.00	1.00	63.96	1.00	1.00	1.00	1.00	1.00
60KB	1.00	1.00	55.55	1.00	1.00	1.00	1.00	1.00

Figure 5.11: SNR Results for the Second Cover Audio

The second cover audio used in this round was longer than the first cover audio used. As a result, each algorithm can embed a higher capacity compared to the previous one. Figure 5.11 highlights the superiority of Indrayani (2020). The BCM-LSB 1 *bps* variation and Bharti et al. (2019) were 0.02 dB lower than Indrayani (2020), while Usanto (2022) was 0.01 dB lower than the same algorithm. These values were resulted from the slightly higher number of flipped bits during the embedding process of each algorithm. Additionally, Alsabhany et al. (2020) continued to show a similar pattern, in which the SNR values were decreasing significantly as it adaptively changed the number of bits embedded per sample depending on the size of secret message.

The SNR values for the third experiment's cover audio are depicted in Figure 5.12. The range sizes of the secret message embedded were the same as the range sizes of the secret message used for the embedding process in Figure 5.11.

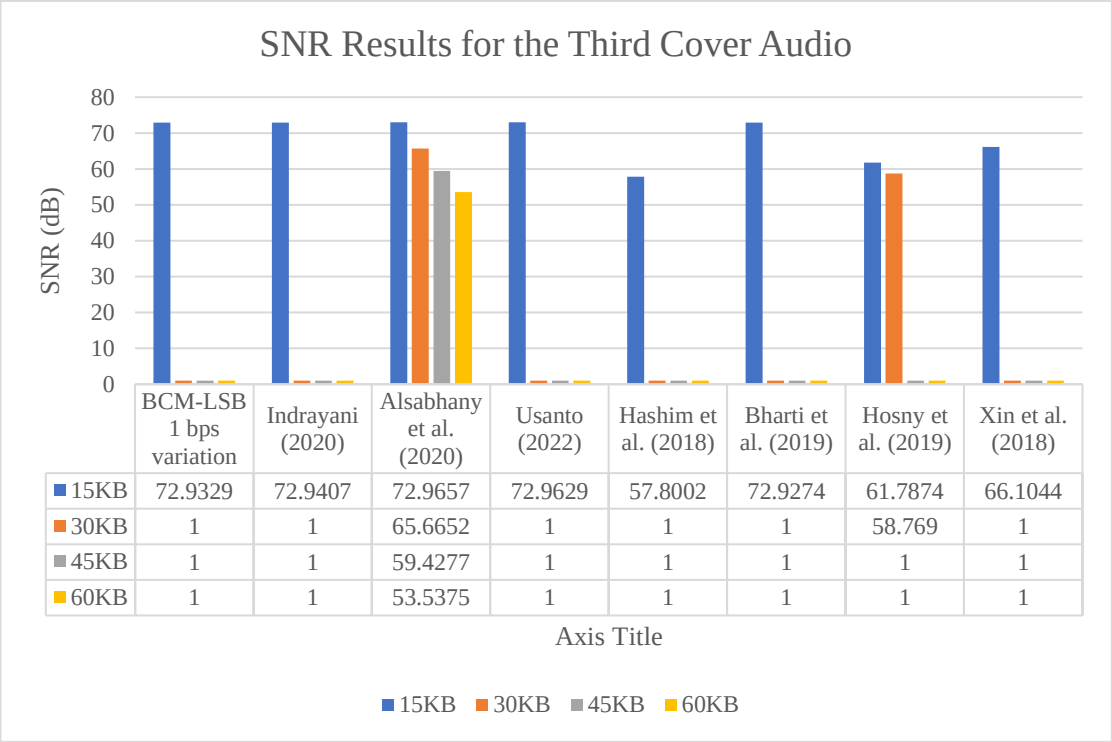


Figure 5.12: SNR Results for the Third Cover Audio

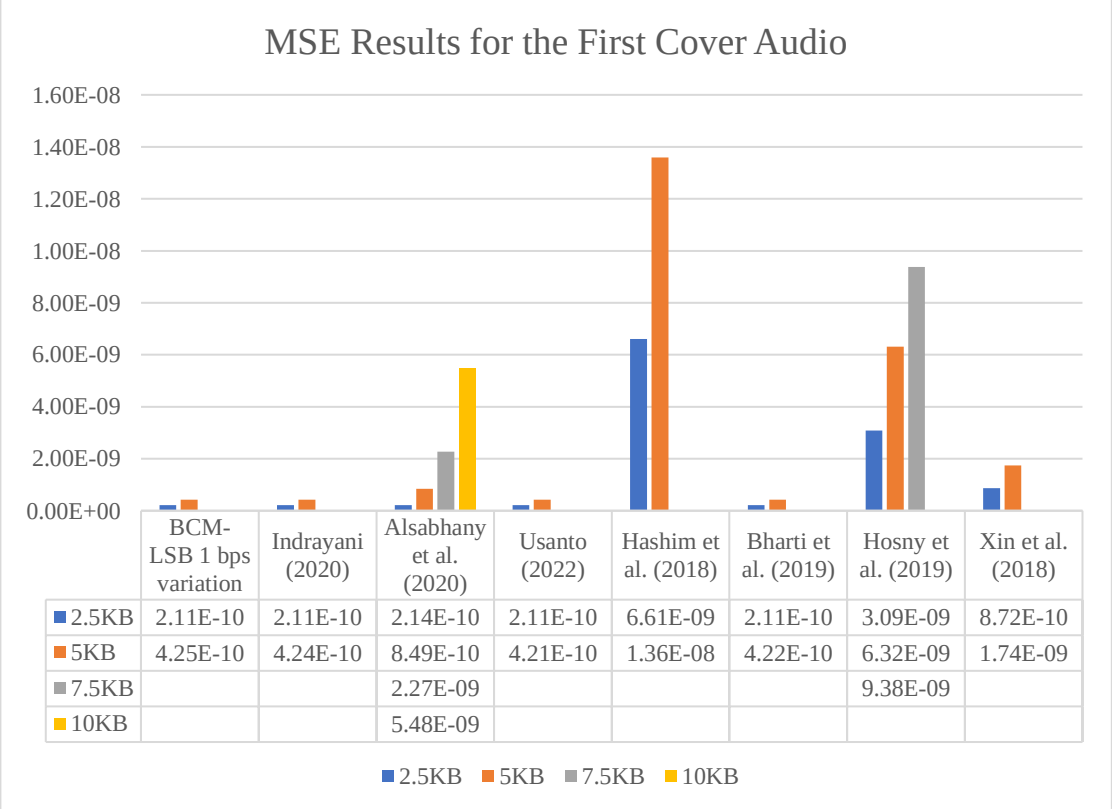
Based on Figure 5.12, Alsabhany et al. (2020) were able to achieve the highest SNR value (72.97 dB) when compared to other algorithms. It produced an SNR value that was 0.04 dB higher than BCM-LSB 1 *bps* variation and Bharti et al. (2019), 0.03 dB higher than Indrayani (2020), and 0.01 dB higher than Usanto (2022). This outcome was caused by the slightly different number of flipped bits occurred throughout the embedding process of each algorithm at the same LSB level.

Another significant finding among across the three cover audio files used was that the second cover was able to attain the highest SNR in a condition that was comparable to that of the other two covers. This is because the second cover has higher audio sample amplitudes than the other two cover files. Based on the SNR formula in Equation 3.1, the higher the amplitude (the numerator in Equation 3.1), the higher the value of SNR.

In summary, the algorithm that embeds at the lowest LSB obtain a higher SNR value in general compared to the algorithm that embeds at a higher level of LSB. In

addition, cover audio files, which yield higher amplitude in general, produce higher SNR values for the same embedding algorithm. The number of flipping bits is also another factor in determining the value of SNR.

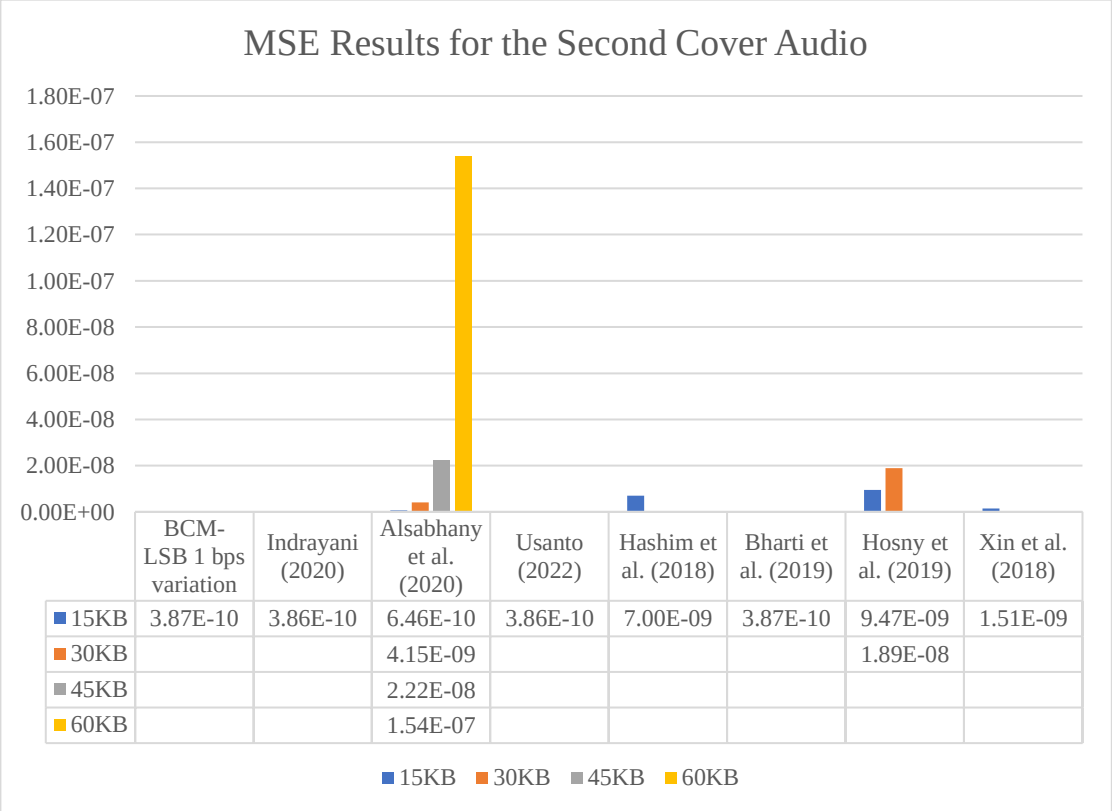
The SNR results represent the most crucial aspect of the comparison experiments. However, additional metrics such as MSE and PSNR must be examined to present a complete picture of the comparison findings, which are used to complement and validate the SNR results for the same cover audio files. In order to prevent duplication, the primary result analysis is already included in the SNR results in the discussion. Figure 5.13 to Figure 5.15 present the MSE values from the embedding result using the first, second and third cover audio files, respectively. The figure table's blank values serve to indicate embedding failure caused by a capacity restriction.



	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
2.5KB	2.11E-10	2.11E-10	2.14E-10	2.11E-10	6.61E-09	2.11E-10	3.09E-09	8.72E-10
5KB	4.25E-10	4.24E-10	8.49E-10	4.21E-10	1.36E-08	4.22E-10	6.32E-09	1.74E-09
7.5KB			2.27E-09				9.38E-09	
10KB			5.48E-09					

Figure 5.13: MSE Results for the First Cover Audio

The high MSE indicates low imperceptibility as reflected in the SNR evaluation. Figure 5.13 shows that algorithm that achieved high SNR values, such as Alsabhany et al. (2020) at 2.5 KB, BCM-LSB 1 *bps* variation, Indrayani (2020), Bharti et al. (2019) and Usanto (2022) generated the lowest MSE. These results confirmed the SNR results for the same sample. The highest MSE was produced by Hashim et al. (2018) at 1.36E-08 because it produced the most distortion to the audio from its embedding process compared to the others. The first cover audio produced the highest MSE compared to the other two cover audio used.

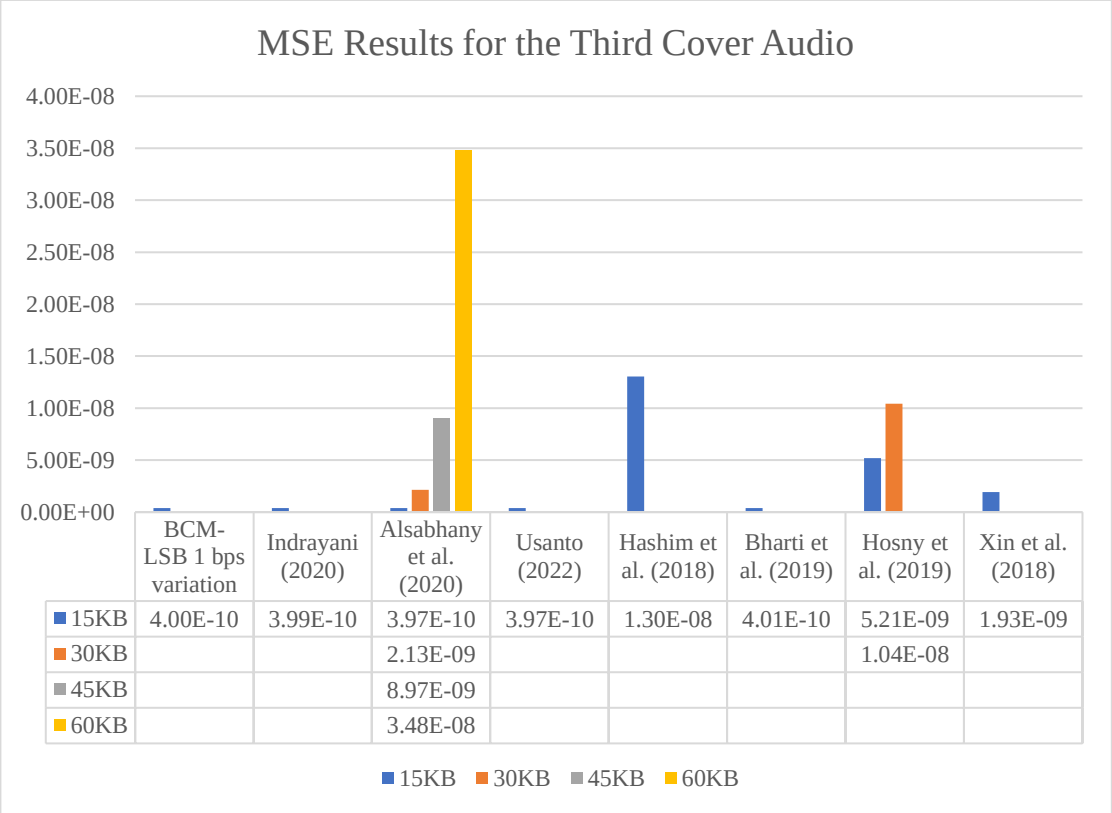


	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
15KB	3.87E-10	3.86E-10	6.46E-10	3.86E-10	7.00E-09	3.87E-10	9.47E-09	1.51E-09
30KB			4.15E-09				1.89E-08	
45KB			2.22E-08					
60KB			1.54E-07					

Figure 5.14: MSE Results for the Second Cover Audio

Figure 5.14 shows a similar result where Alsabhany et al. (2020) at 15KB, BCM-LSB 1 *bps* variation, Indrayani (2020), Bharti et al. (2019) and Usanto (2022) generated the lowest MSE. However, Alsabhany et al. (2020) produced the highest MSE as it embedded 60 KB of secret message hence produced the most distortion output from the

embedding process. On the other hand, Hosny et al. (2019) produced the highest MSE when embedded 15 KB of secret message compared to the other algorithms.

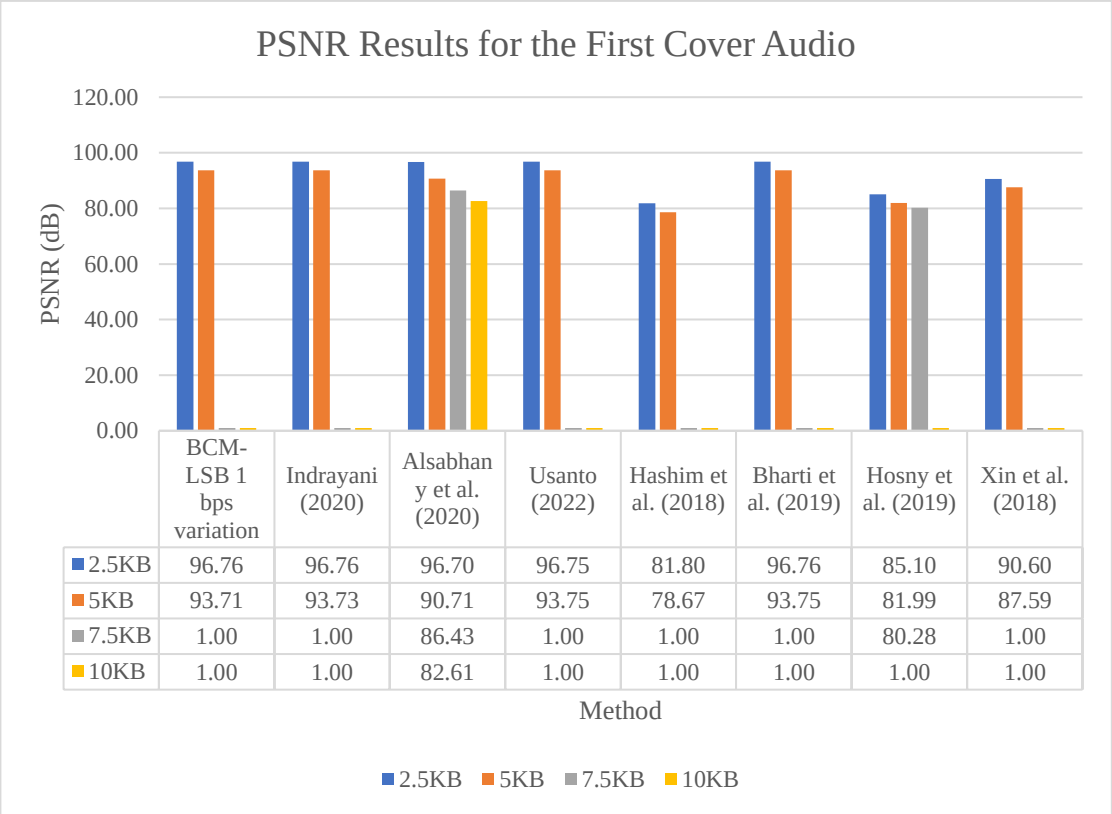


	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
15KB	4.00E-10	3.99E-10	3.97E-10	3.97E-10	1.30E-08	4.01E-10	5.21E-09	1.93E-09
30KB			2.13E-09				1.04E-08	
45KB			8.97E-09					
60KB			3.48E-08					

Figure 5.15: MSE Results for the Third Cover Audio

Figure 5.15 shows that among from the lowest MSE values were achieved by Alsabhany et al. (2020), BCM-LSB 1 *bps* variation, Indrayani (2020), Bharti et al. (2019) and Usanto (2022). At 15KB, Hashim et al. (2018) produced the highest MSE.

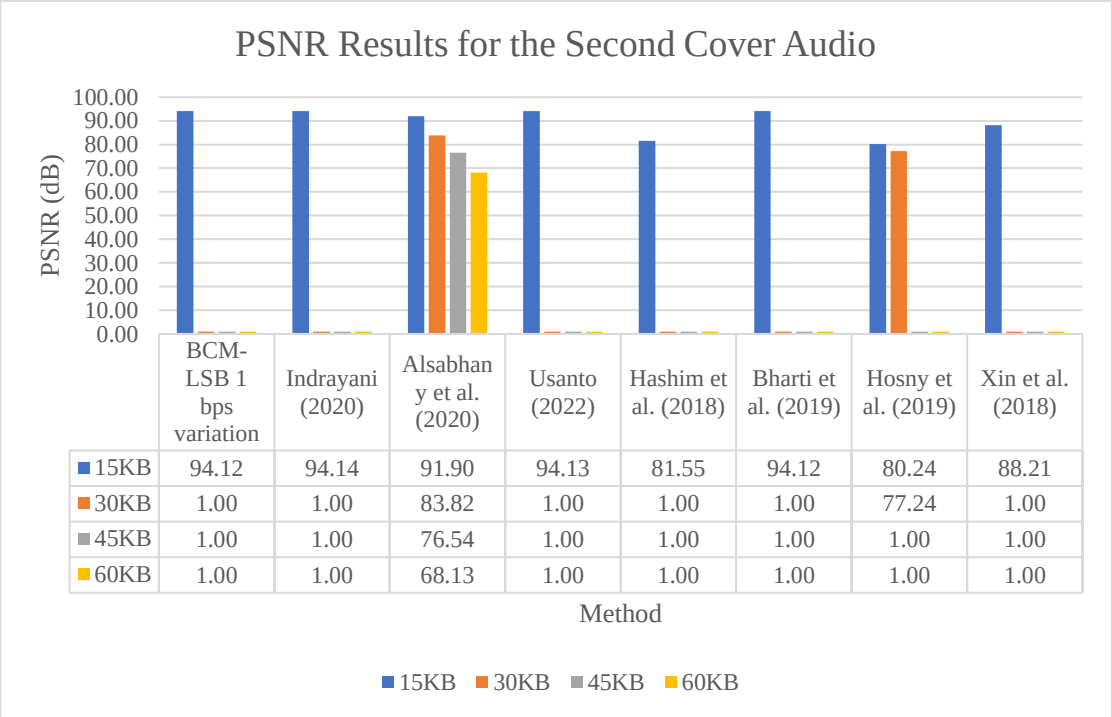
Figures 5.16 to Figure 5.18 show the PSNR values for the first, second, and third cover audio files in this comparison experiment, respectively. A value of 1 is used to indicate the embedding failure cases.



	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
2.5KB	96.76	96.76	96.70	96.75	81.80	96.76	85.10	90.60
5KB	93.71	93.73	90.71	93.75	78.67	93.75	81.99	87.59
7.5KB	1.00	1.00	86.43	1.00	1.00	1.00	80.28	1.00
10KB	1.00	1.00	82.61	1.00	1.00	1.00	1.00	1.00

Figure 5.16: PSNR Results for the First Cover Audio

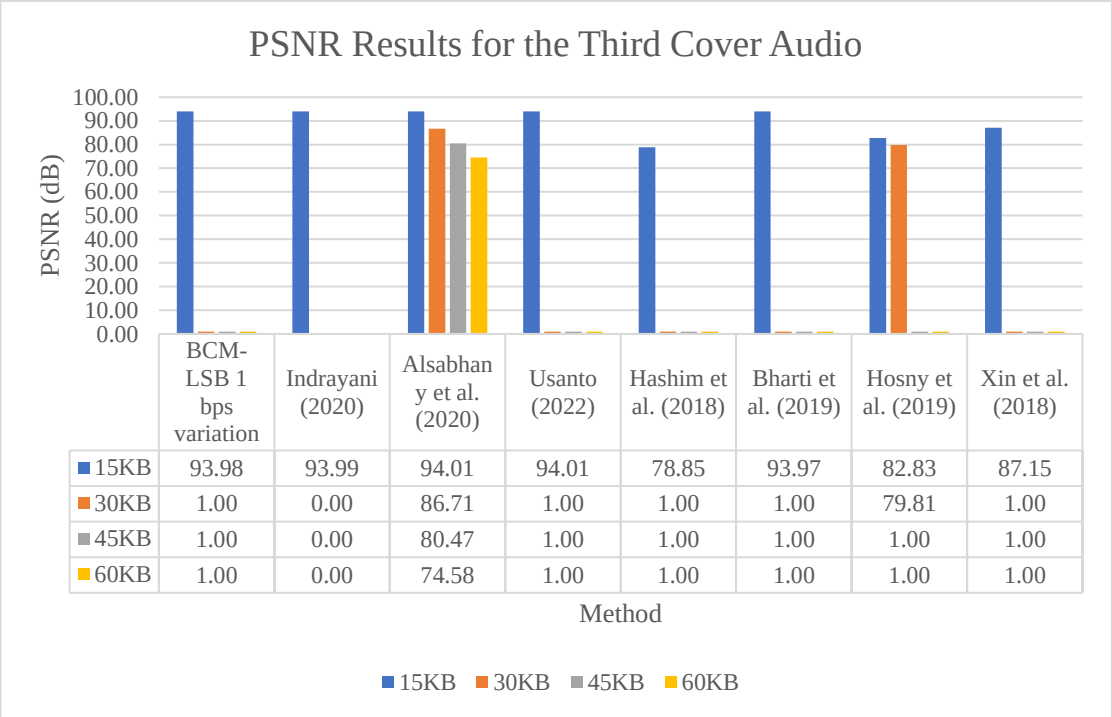
The PSNR values are in sync with the SNR results. However, the PSNR values are more error tolerant than SNR as they measure the highest possible amplitude for each sample instead of taking the current amplitude value. Based on Figure 5.16, the highest PSNR values were achieved by BCM-LSB 1 *bps* variation, Indrayani (2020), Bharti et al. (2019) followed by Usanto (2022) and Alsabhany et al. (2020). The lowest PSNR recorded at 78.67 dB by Hashim et. al (2018) after embedding 5 KB of secret message.



	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
15KB	94.12	94.14	91.90	94.13	81.55	94.12	80.24	88.21
30KB	1.00	1.00	83.82	1.00	1.00	1.00	77.24	1.00
45KB	1.00	1.00	76.54	1.00	1.00	1.00	1.00	1.00
60KB	1.00	1.00	68.13	1.00	1.00	1.00	1.00	1.00

Figure 5.17: PSNR Results for the Second Cover Audio

Based on Figure 5.17, the algorithms that embed at the lowest LSB which are BCM-LSB 1 *bps* variation, Indrayani (2020), Bharti et al. (2019) and Usanto (2022) have the highest PSNR value compared to the algorithms that embed at the higher level of LSB. Alsabhany et al. (2020) has slightly lower PSNR at 15 KB which indicates that it embeds some of secret message at slightly higher LSB. Alsabhany et al. (2020) also produced the highest PSNR as this algorithm alone managed to embed with bigger capacity, which logically led to a bigger distortion.



	BCM-LSB 1 bps variation	Indrayani (2020)	Alsabhany et al. (2020)	Usanto (2022)	Hashim et al. (2018)	Bharti et al. (2019)	Hosny et al. (2019)	Xin et al. (2018)
15KB	93.98	93.99	94.01	94.01	78.85	93.97	82.83	87.15
30KB	1.00	0.00	86.71	1.00	1.00	1.00	79.81	1.00
45KB	1.00	0.00	80.47	1.00	1.00	1.00	1.00	1.00
60KB	1.00	0.00	74.58	1.00	1.00	1.00	1.00	1.00

Figure 5.18: PSNR Results for the Third Cover Audio

Based on Figure 5.18, the PSNR values produced similar analysis found in Figures 5.16 and Figure 5.17. The highest PSNR values were achieved by BCM-LSB 1 *bps* variation, Indrayani (2020), Bharti et al. (2019), and Usanto (2022) and sometimes by Alsabhany et al. (2020), when this algorithm embedded at the lowest LSB.

In summary, the algorithms that embed at the lowest LSB have the highest imperceptibility characteristic, while the algorithms that embed at the highest LSB tend to have the lowest imperceptibility characteristic when compared among audio steganography LSB techniques. It is because embedding at the lowest LSB modified the audio samples and produced the lowest difference in values between the stego-file and cover file. This finding is one of the justifications to keep using low *bps* variations in the proposed algorithm. However, the capacities of these variations are low, which are demonstrated previously during parameter setting. BCM-LSB aims to capture all the best performance on each characteristic to ensure the user can choose the most

suitable one depending on the situation. In the next section, the capacity of BCM-LSB is evaluated.

5.3.1.2 Capacity Characteristic Performance Comparison Results

The objective of this evaluation was to compare the BCM-LSB algorithm's capacity against existing algorithms in terms of maximum size of secret message that can be embedded or known as maximum capacity. The comparison considers three cover audio signals of various length and contents. The capacity of each algorithm is presented in Table 5.13.

Table 5.13: The Maximum Capacity Values for Three Cover Audio Files

Algorithm	***Maximum Capacity (KB)***		
	First Cover Audio	**Second Cover Audio**	**Third Cover Audio**
8 *bps* variation	44.056	141.0240	140.12
Indrayani (2020)	5.507	17.6280	17.515
Alsabhany et al. (2020)	20.5058	69.4185	98.718
Usanto (2022)	5.507	17.6280	17.515
Hashim et. al. (2018)	5.507	17.6280	17.515
Bharti et. al. (2019)	5.507	17.6280	17.515
Hosny et. al. (2019)	8.1918	32.7678	32.7678
Xin et. al. (2018)	5.5070	17.6285	17.5152
Min	5.507	17.628	17.515
Max	44.056	141.024	140.12

Table 5.14 shows the 8 *bps* variation tops other algorithms by a high margin. Indrayani (2020), Bharti et al. (2019), Hashim et al. (2018) and Usanto (2022) produced the same lowest maximum capacity as all of them embed 1 *bps*, although based on the

imperceptibility evaluation previously, they have different imperceptibility performance due to the difference of LSB level that the embedding takes place. Xin et al. (2018) also produced the lowest capacity however the reason was different. Although this algorithm embeds as high as 3 *bps*, it depends on the value of the amplitude. This algorithm sets two thresholds and only if the amplitude is higher than either one or both thresholds, then it embeds two and three bits per sample, respectively. In all three cover audio files used, there were not many audio samples' amplitude that passed the thresholds. It is also worth noticing that Hosny et al. (2019) had a similar maximum capacity between second cover audio and third cover audio. It is because its maximum capacity is determined by the Linear-feedback shift register based on the Feedback polynomial. For the second and third cover audio the maximum capacity was determined by using $x^{17} + x^{14} + 1$ which produced 131,071 sequences, while on the other hand, the maximum capacity for the first cover audio was determined by using $x^{15} + x^{14} + 1$, which is 32,767 sequences. Since it embeds 2 bits per sample, the number of sequences is multiplied by 2 which is then divided by 8 to get the capacity in bytes instead of bits.

In summary, the algorithms that embed at the highest LSB have the highest capacity characteristic, while the algorithms that embed at the lowest LSB tend to have the lowest capacity characteristic when compared among audio steganography LSB techniques. This finding is one of the reasons that inspired the keeping of the high *bps* variations in the proposed algorithm. However, the imperceptibility of variation is low, which are demonstrated previously. BCM-LSB aims to capture all the best performance on each characteristic to ensure the user can choose the most suitable one depending on the situation. In the next section, the robustness of BCM-LSB is evaluated.

5.3.1.3 Robustness Characteristic Performance Comparison Results

The objective of this evaluation was to compare the BCM-LSB algorithm against existing algorithms in terms of robustness against AWGN attack using the first, second, and third cover audio. The experiment demonstrates the resistance of the proposed algorithm and related algorithms against various loads of AWGN. In this experiment, half of each algorithm's maximal capacity was embedded. The recovery procedure is then executed by utilising the recovery portion of each technique to locate the embedded message. The performance of the offered techniques is captured using the three prior cover audios. Figure 5.19 depicts the effects of the AWGN attack on the first cover audio file.

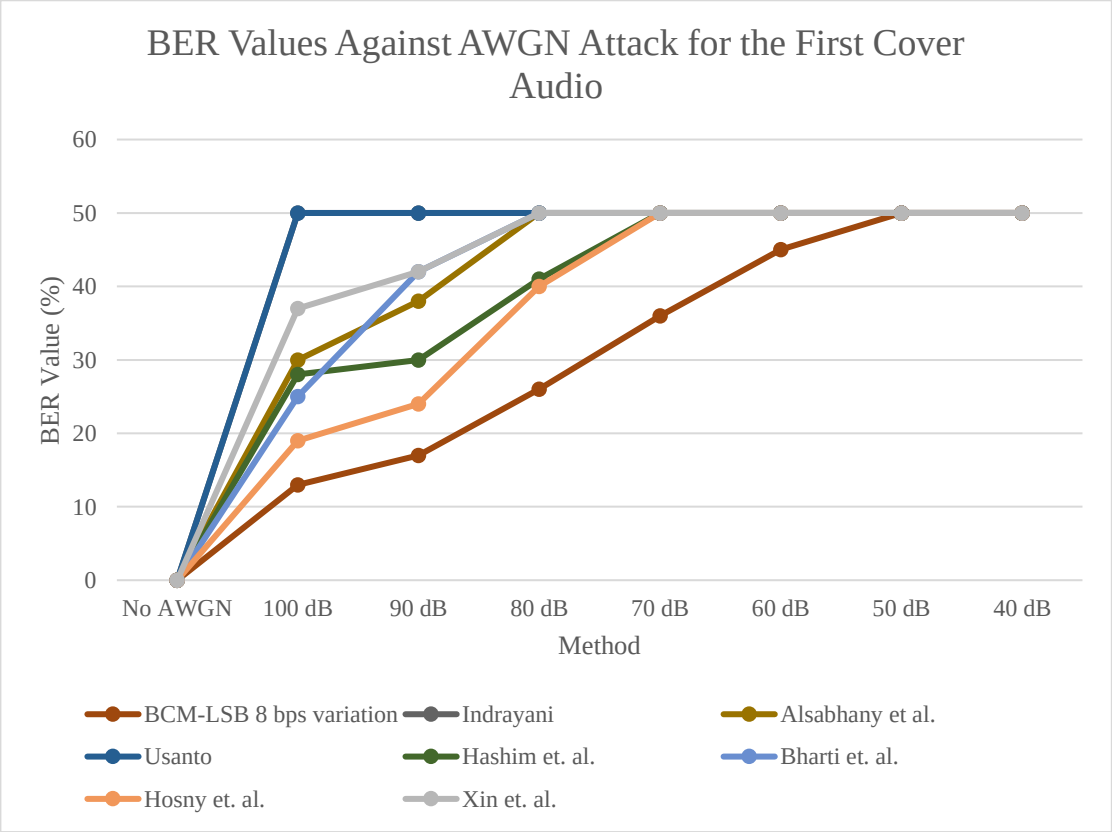


Figure 5.19: BER Values Against AWGN Attack for First Cover Audio

Figure 5.19 highlights the lower BER values produced by the BCM-LSB variations against all the other algorithms for all AWGN levels excepts at 50 dB and 40 dB. Specifically, the BCM-LSB 8 *bps* variation achieved the lowest bit error rates among other algorithms, especially at the 100 dB attack level. In general, the higher the LSB level used for embedding, the lower the error rate produced by the attack. All algorithms reached the entire loss point at 60 dB, which happens when the BER exceeds 45% (Djebbar et al., 2010), indicating that the validity of the restored bits is subject to random chance, namely 50%. In summary, it is known that stego-file that produced from the embedding process at higher LSB has higher resistance towards the attack (Hosny et al., 2019) hence BCM-LSB that embeds as high as eighth level has the highest resistance. Next, Figure 5.20 depicts the effects of the AWGN attack on the second cover audio file.

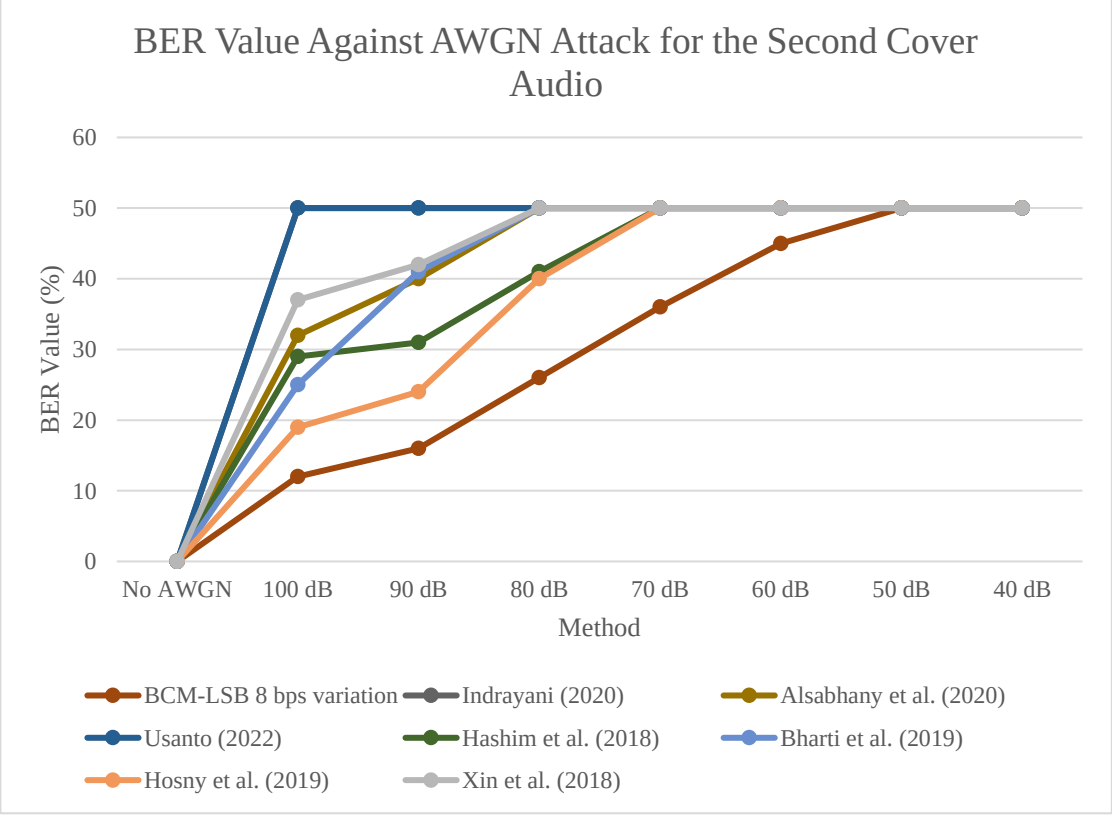


Figure 5.20: BER Values Against AWGN Attack for Second Cover Audio

Based on Figure 5.20, the BCM-LSB achieved better noise resistance compared to other algorithms. Other algorithms could not withstand the AWGN attack at 80 dB except Hashim et.al. (2018) and Hosny et al. (2019) while BCM-LSB 8 *bps* were able to withstand another 10dB attack which was at 70dB. The results of the AWGN attack for the third sample are shown in Figure 5.21.

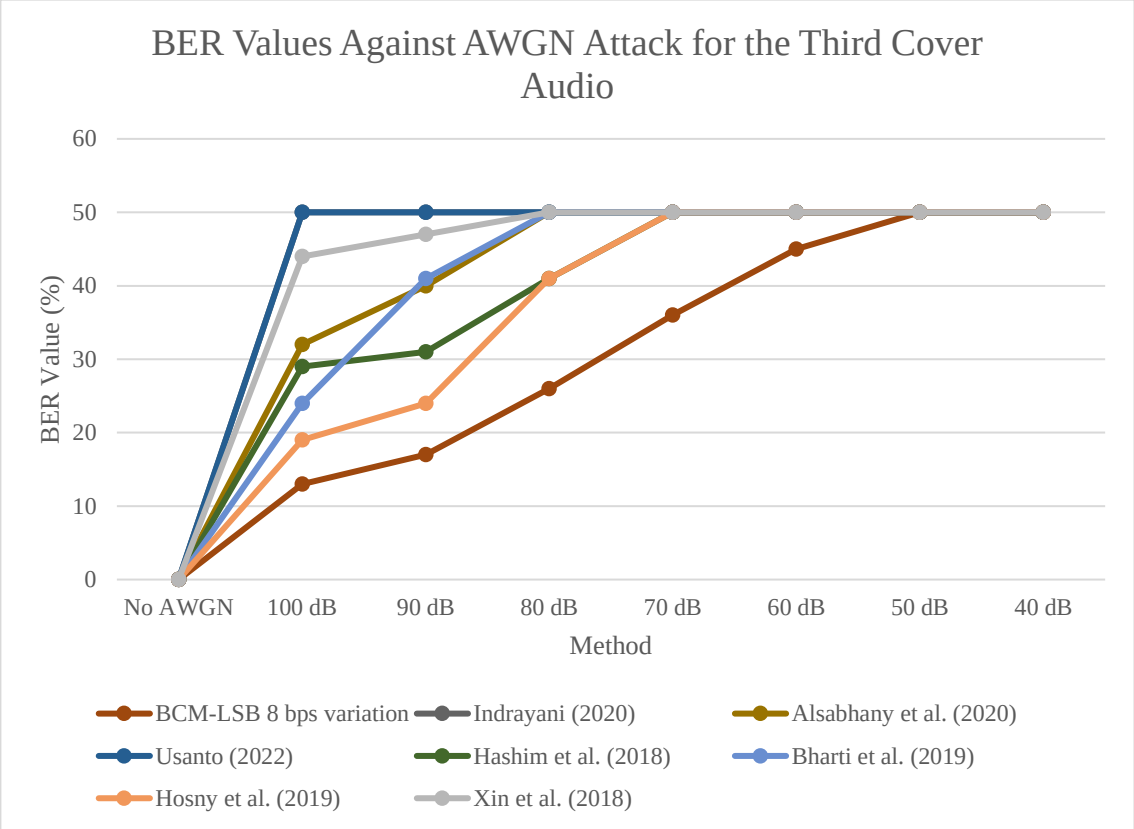


Figure 5.21: BER Values Against AWGN Attack for Third Cover Audio

Based on Figure 5.21, the BCM-LSB 8 *bps* variation achieved the lowest BER amongst all the levels and therefore showed the highest robustness compared to the other algorithms. Hosny et al. (2019)'s robustness is second best among all algorithms since it embeds at a slightly lower level of LSB than BCM-LSB.

In summary, BCM-LSB 8 *bps* achieves the highest robustness compared to other algorithms. In general, the higher the level of LSB, which is picked for the embedding

process, the higher the resistance toward the AWGN attack. In addition, generally, all algorithms cannot withstand 60 dB, 50 dB and 40 dB level of AWGN attack because the LSB embedding technique has low robustness compared to other audio steganography techniques such as phase coding. Lastly, the BER values slightly increased or decreased when compared to the same attack and algorithms across the cover audio files used. This is because the parts of the audio file with greater changes in amplitude can outperform the silent intervals of audio files (Ahmed A Alsabhany, 2019).

5.3.1.4 Dynamic Security Characteristic Performance Comparison Results

The objective of this evaluation was to compare the BCM-LSB algorithm's dynamic security against existing algorithms using three audio signals of various length and content. The first test was conducted using the Seg-SNR Spike Test and the second one was using the Difference Signal Test. The *key x, key r* and *key n* were set at 0.65, 3.95 and 1024 respectively.

Seg-SNR Spike Test was conducted in three rounds using three different cover audios similarly used in the previous imperceptibility comparison experiment. The first round was conducted using the first cover audio file. Figure 5.22 and Figure 5.23 present the seg-SNR spikes at 25% and 35% independent embedding capacities, respectively.

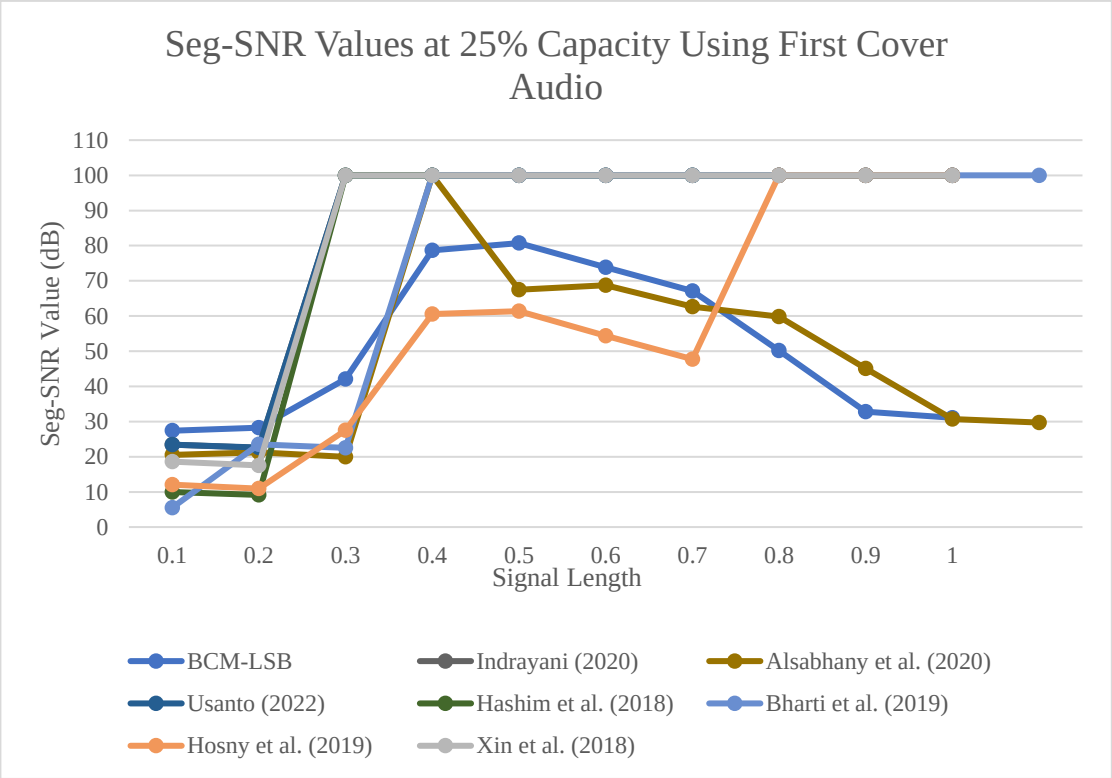


Figure 5.22: Seg-SNR Values at 25% Capacity Using First Cover Audio

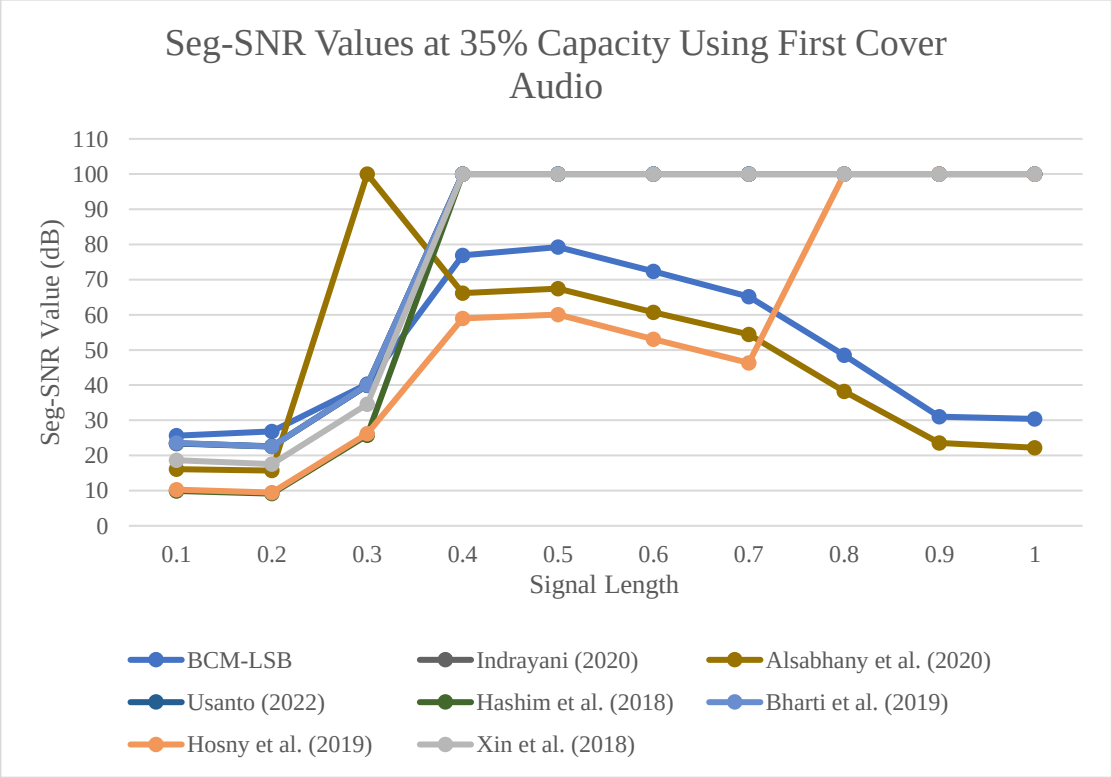


Figure 5.23: Seg-SNR Values at 35% Capacity Using First Cover Audio

Based on Figure 5.22, all algorithms managed to maintain the quality of the first three-tenths of the cover audio file. However, a significant spike to infinity then took place at 30% of the cover audio for Xin et al. (2018), Bharti et al. (2019), Indrayani (2020), Usanto (2022) and Hashim et al. (2018). This situation occurred due to their sequential embedding behavior, indicating their direct dependence on fulfilling the maximum capacity as per the cover audio. On the other hand, Alsabhany et al. (2020) and Hosny et al. (2019) managed to spread the secret message better compared to the others as Hosny et al. (2019) managed to spread up the secret message until 70% of the cover audio, while Alsabhany et al. (2020) managed to spread throughout the entire cover audio except between 30% and 40% capacity.

A similar pattern was shown in Figure 5.23 by other related algorithms. Xin et al. (2018), Bharti et al. (2019), Indrayani (2020), Usanto (2022) and Hashim et al. (2018) were limited by the sequential way of embedding, which reduced their dynamic security. Alsabhany et al. (2020) missed embedding 30% - 40% of audio segment due to its threshold-based embedding nature. Hosny et al. (2019), on the other hand, cannot embed the secret message after 70% part of the cover audio as the maximum capacity was dependent on the maximum value of Linear-feedback Shift Register which is lower than the total audio samples of the cover audio used.

BCM-LSB was able to successfully embed the secret message within the cover audio file, avoiding a spike to infinity. The large fluctuations in the seg-SNR can be attributed to the significant differences in the original amplitudes of each segment.

Next, the second round was conducted using the second cover audio file. Figure 5.24 and Figure 5.25 present the seg-SNR spikes at 25% and 35% independent embedding capacities, respectively.

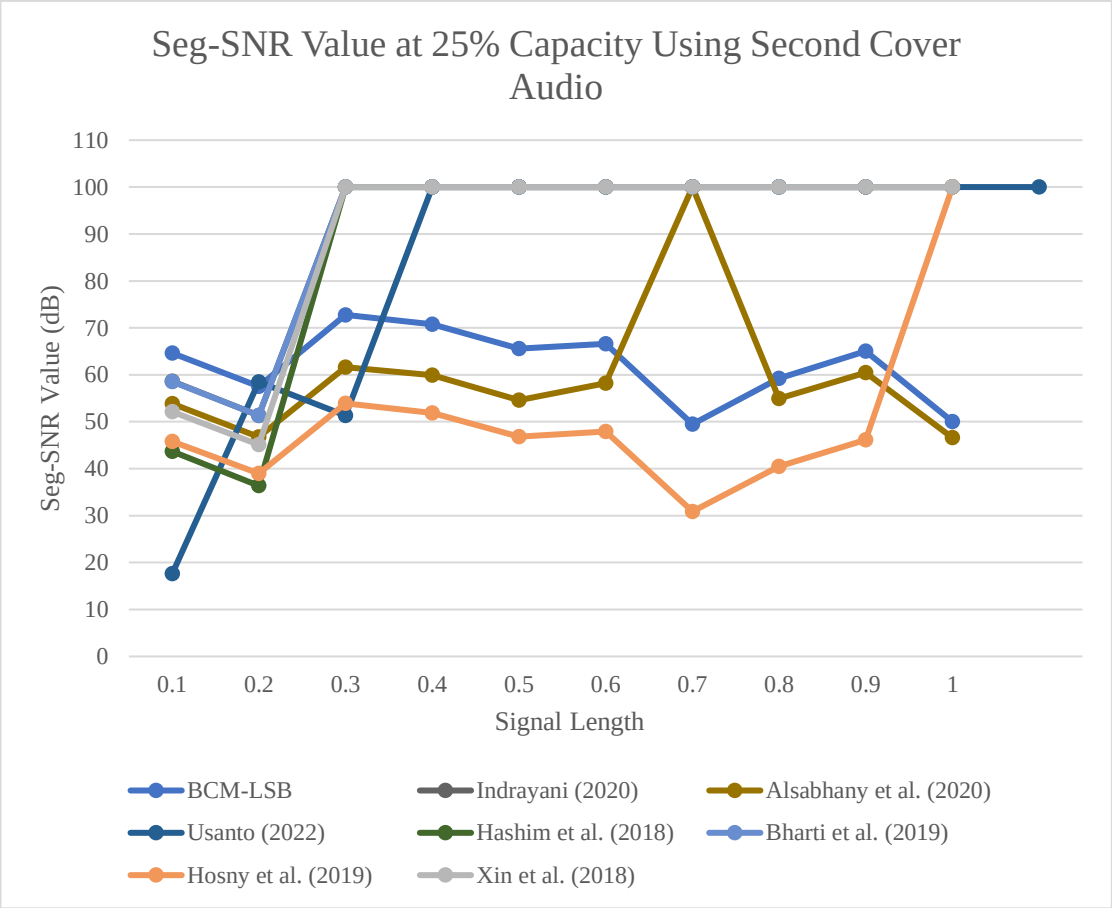


Figure 5.24: Seg-SNR Values at 25% Capacity Using Second Cover Audio

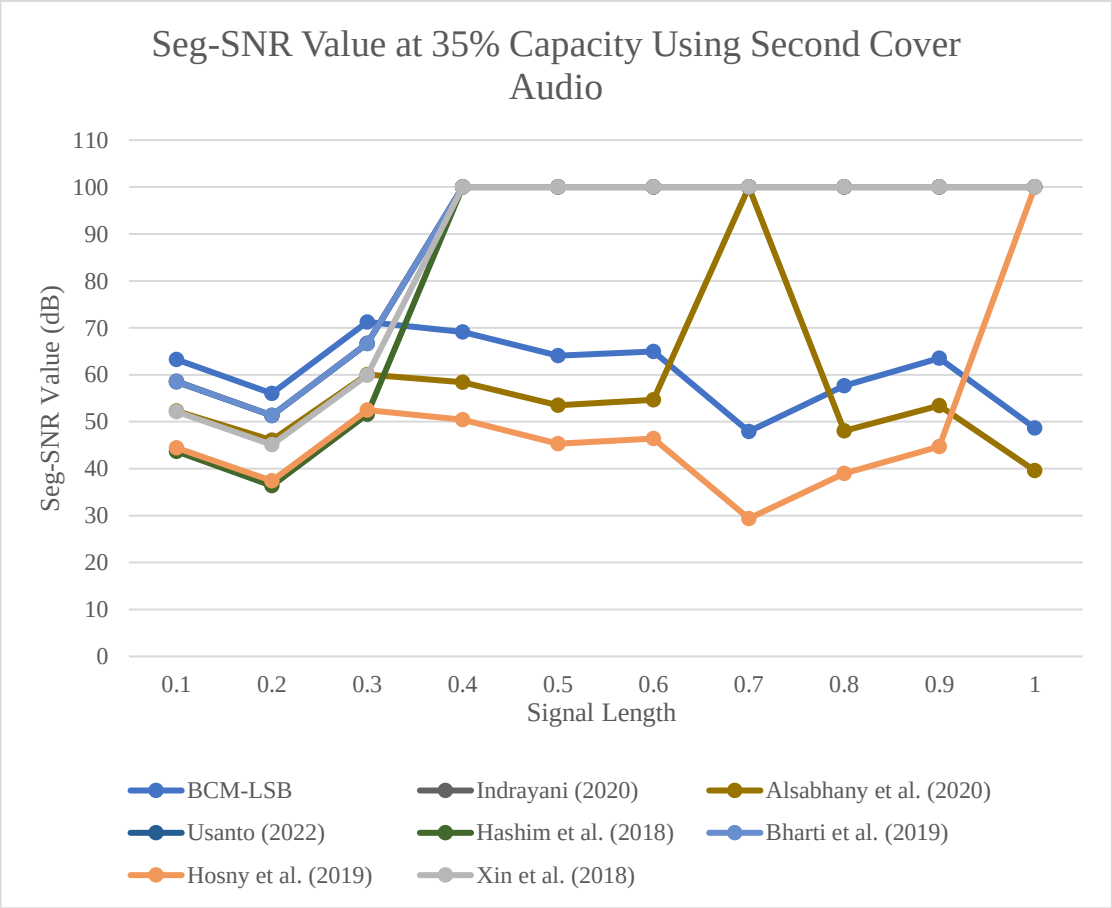


Figure 5.25: Seg-SNR Values at 35% Capacity Using Second Cover Audio

Based on Figure 5.24 and Figure 5.25, a similar pattern emerged in all related sequential embedding behaviour. Significant spiked to infinity took place at 30% and 40% of the cover audio in Figure 5.22 and Figure 5.23, respectively, for Xin et al. (2018), Bharti et al. (2019), Indrayani (2020), Usanto (2022) and Hashim et al. (2018). Alsabhany et al. (2020) and Hosny et al. (2019) were found to be more successful in spreading the secret message compared to others. Hosny et al. (2019) was able to spread the message until 90% of the cover audio, while Alsabhany et al. (2020) was able to spread the message throughout the entire cover audio, except for a small segment at 70%. BCM-LSB do not produce any sudden spike to infinity, which indicated that they managed to distribute the secret message throughout the second cover audio.

Next, the third round was conducted using the third cover audio file. Figure 5.26 and Figure 5.27 present the seg-SNR spikes at 25% and 35% independent embedding capacities, respectively.

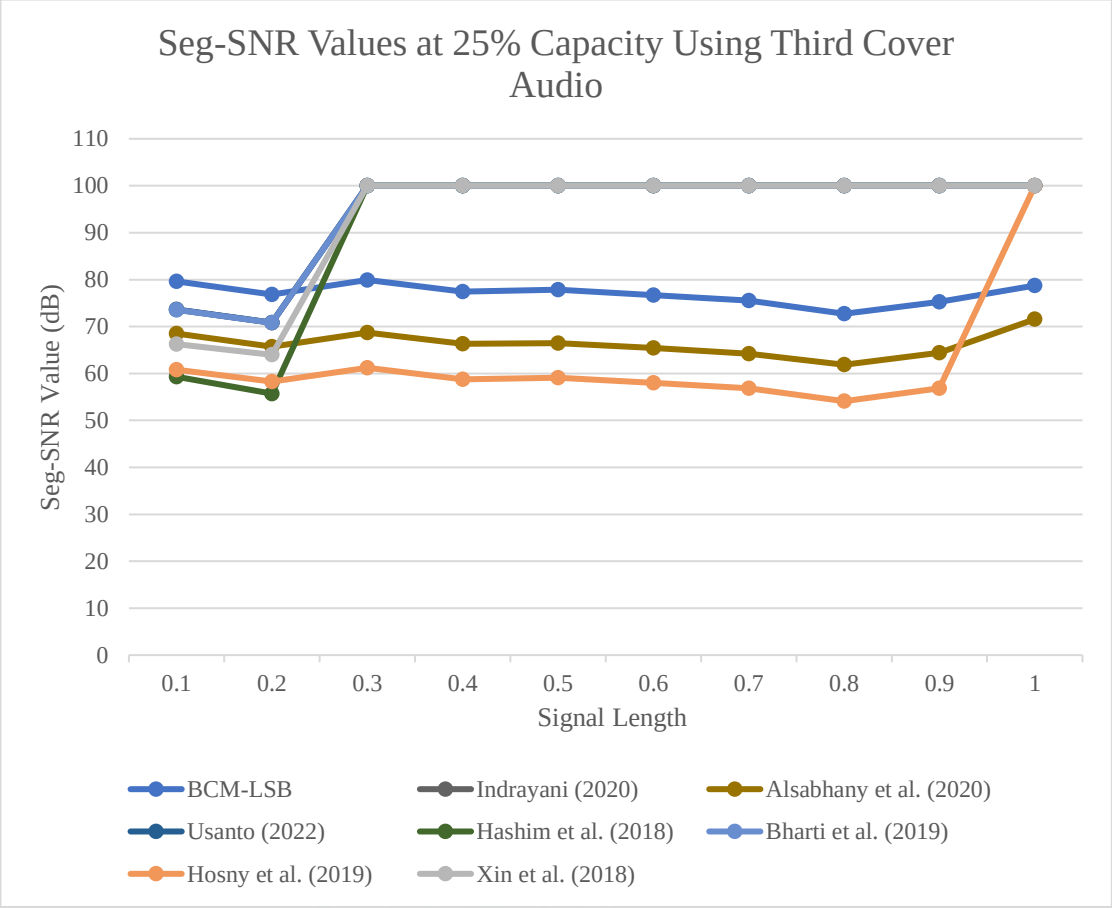


Figure 5.26: Seg-SNR Values at 25% Capacity Using Third Cover Audio

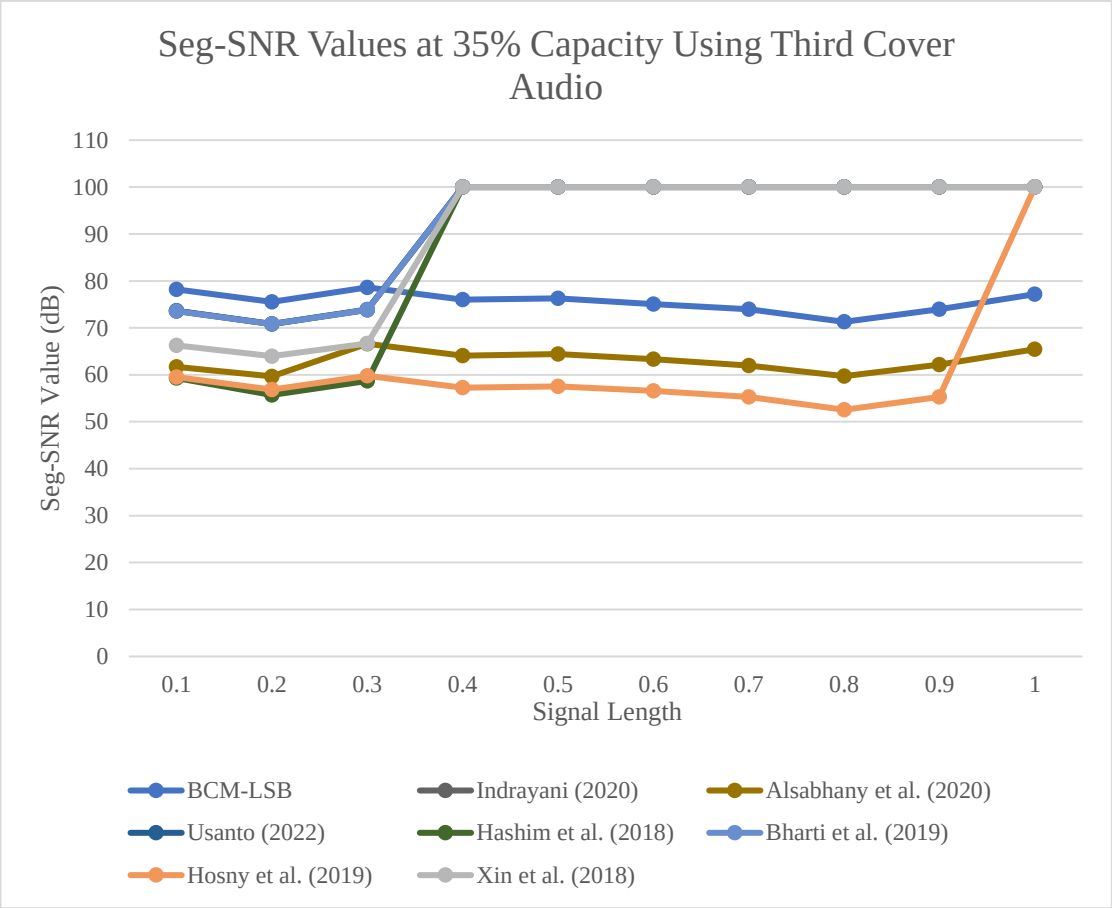


Figure 5.27: Seg-SNR Values at 35% Capacity Using Third Cover Audio

Based on Figure 5.26 and Figure 5.27, it can be observed that the seg-SNR results for the third cover audio differ from those of the previous cover audios. Xin et al. (2018), Bharti et al. (2019), Indrayani (2020), Usanto (2022) and Hashim et al. (2018) still produced a significant spike to infinity at 30% and 40% of cover audio. Similar when using the second cover audio, Hosny et al. (2019) was able to spread the secret message up to 90% of the third cover audio. BCM-LSB 1 *bps* variation and Alsabhany et al. (2020) managed to spread the secret message to the entire cover audio in third cover, compared with result produced from first and second cover, whereas only BCM-LSB algorithm managed to spread the secret message to the entire cover audio. Alsabhany et al. (2020) require minimal audio sample skipping during the embedding process due to the consistent high amplitudes of the audio samples across all covers utilized. This

guarantees an even distribution of the secret message throughout the cover audio. In summary, BCM-LSB has the best dynamic security based on this evaluation followed by Alsabhany et al. (2020) based on the ability to spread secret message throughout the cover. However, Alsabhany et al. (2020) algorithm's nature which takes various low bit embedding and dependant on the audio sample's amplitude value causes inconsistency on the distribution of secret message for different cover audios.

Next, Difference Signal Test was conducted in three rounds with the same setup as previously discussed in Seg-SNR Spike Visual Test. Figure 5.28, (a) to (c) presents the example of the original cover audio file, stego-file and the difference between these two files in wavelet representation. The difference was calculated by subtracting each of stego-file audio sample's value with the cover audio sample's value.

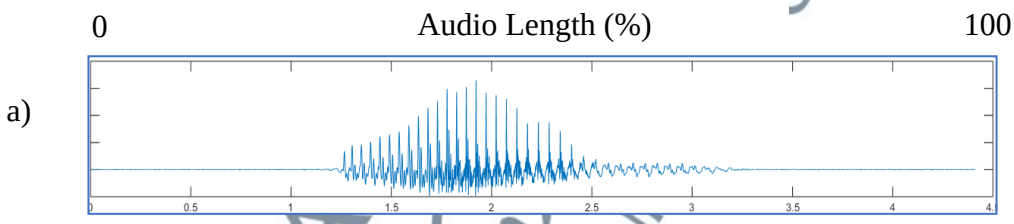


Figure 5.28 (a): Cover Audio File Signal

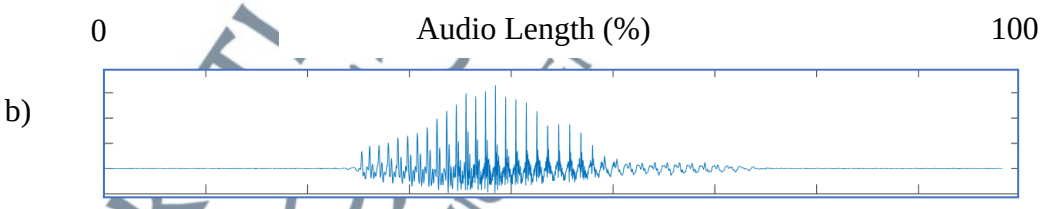


Figure 5.28 (b): Stego-file Signal

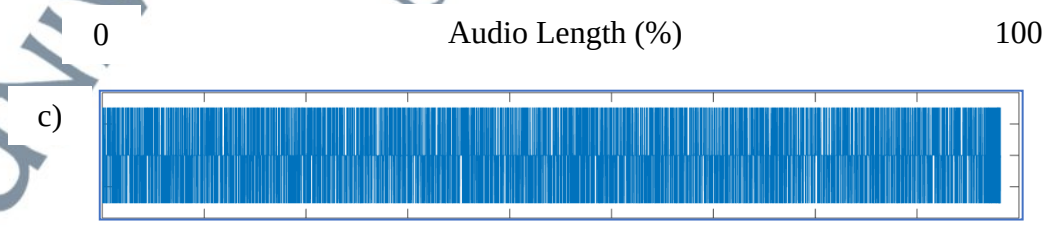


Figure 5.28 (c): Difference Signal

In Figure 5.28, BCM-LSB 1 *bps* variation was used at 25% of its capacity. This figure does not report any analysis value but only serves as an example of how each difference signal is analysed.

The experimental results are presented in Figure 5.29 to Figure 5.31, (a) to (h) for BCM-LSB, Indrayani (2020), Alsabhany et al. (2020), Usanto (2022), Hashim et al. (2018), Bharti et al. (2019), Hosny et al. (2019) and Xin et al. (2018) respectively. The main objective of this experiment is to show the relationship between ways of embedding against capacity percentage. Figure 5.29 shows the signal differences for the first cover audio file.

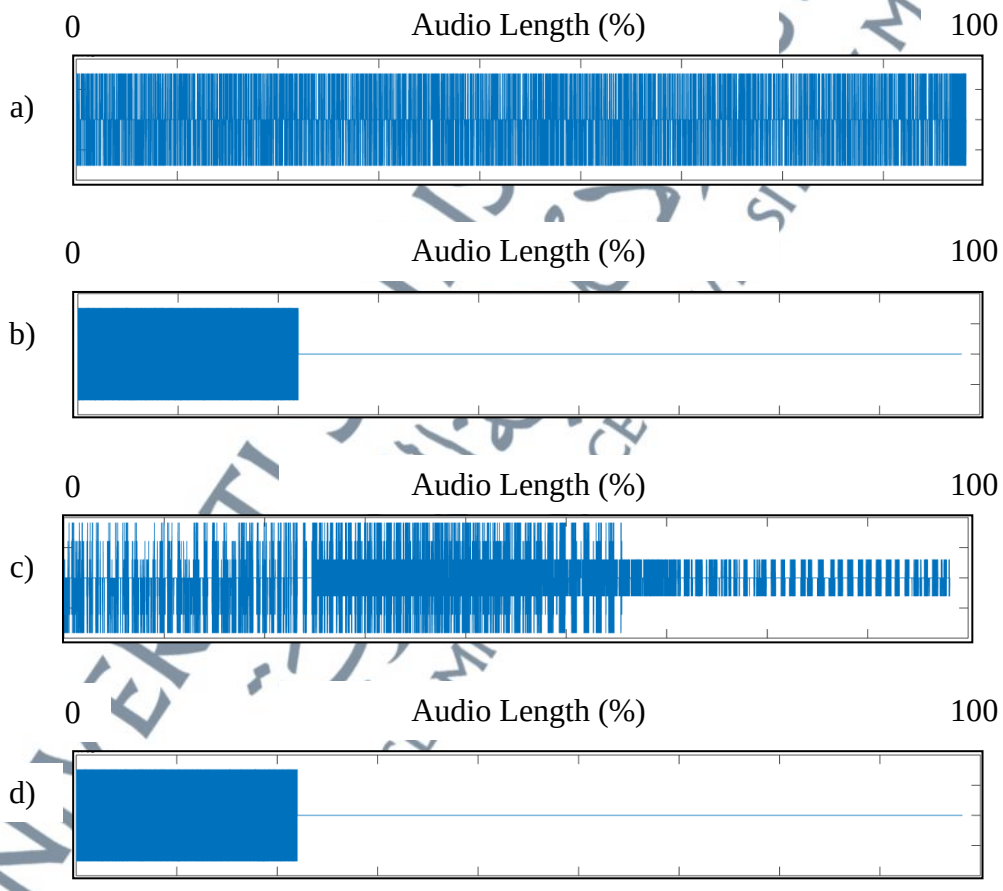


Figure 5.29: Difference Signals at 25% Capacity using First Cover Audio File

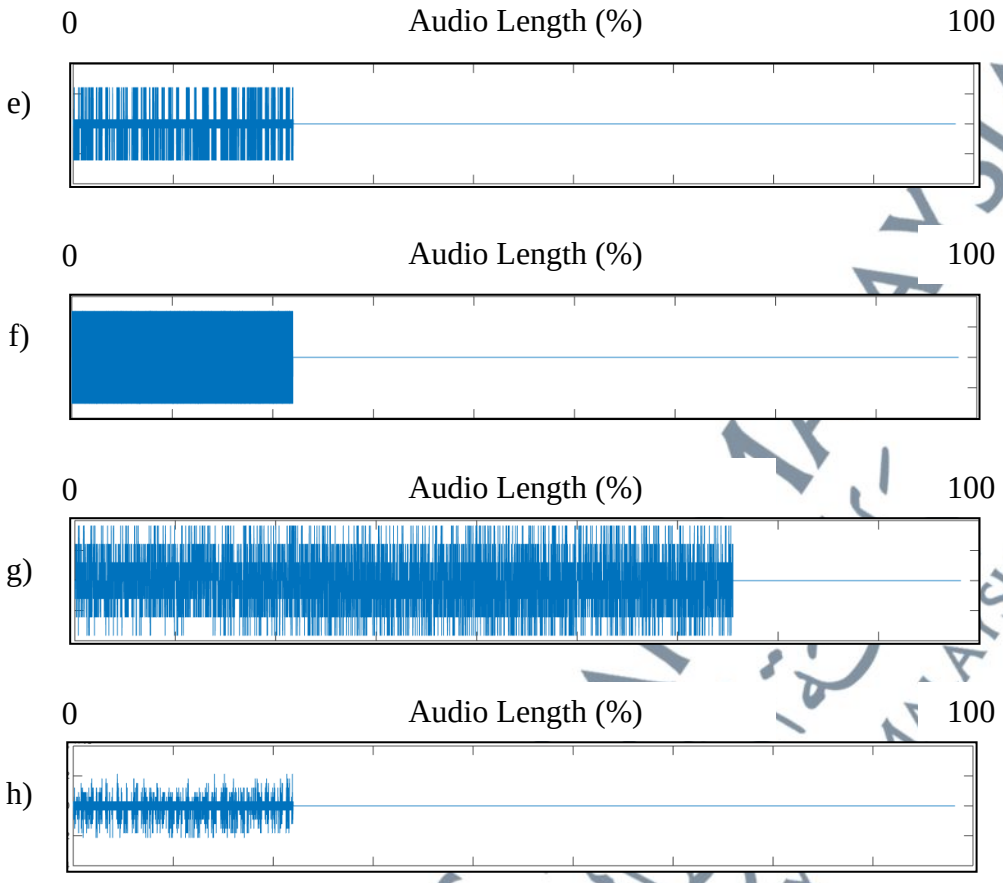


Figure 5.29: Difference Signals at 25% Capacity using First Cover Audio File, continued

Based on Figure 5.29, several trends can be observed. For the algorithms which have sequential embedding behaviours, such as Indrayani (2020), Usanto (2022), Hashim et al. (2018), Bharti et al. (2019) and Xin et al. (2018), they failed to distribute the secret message as there were obvious differences between modified and clean audio samples. Hosny et al. (2019) managed to distribute the secret message until around 70% of the cover audio. On the other hand, although Alsabhany et al. (2020) fairly distributed the secret message until the end of the cover audio, there were huge differences between certain parts of difference signal produced. It was because two or more different LSB levels were used for embedding the secret message. In addition, there was also a small gap at around 30% of the signal difference which indicates this algorithm skipped this part of the embedding process. Meanwhile, BCM-LSB managed to distribute quite

fairly without producing any huge differences between parts of the signal difference. Figure 5.30 shows the signal differences for the second cover audio file.

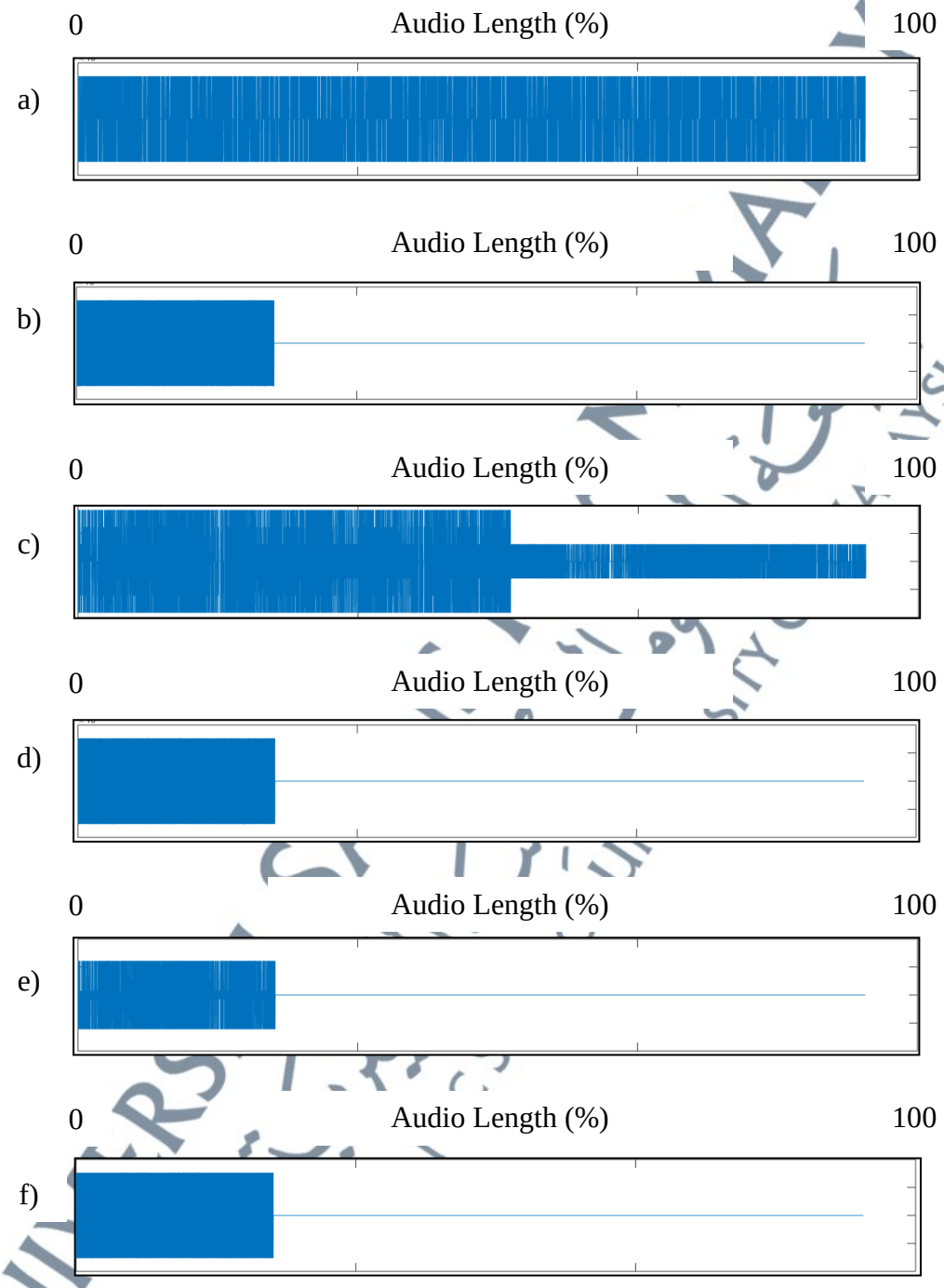


Figure 5.30: Difference Signals at 25% Capacity using Second Cover Audio File

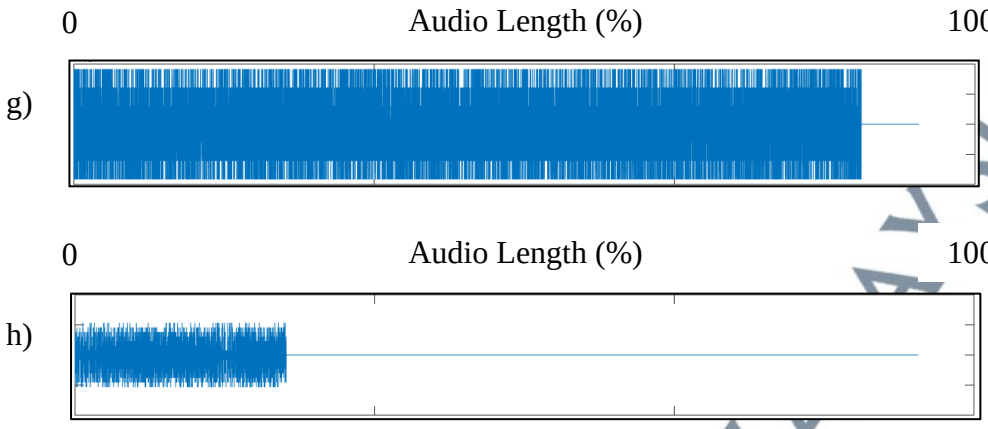


Figure 5.30: Difference Signals at 25% Capacity using Second Cover Audio File, continued

The trend presented in Figure 5.30 is similar to the one shown in Figure 5.29. Indrayani (2020), Usanto (2022), Hashim et al. (2018), Bharti et al. (2019) and Xin et al. (2018) only managed to distribute the secret message to around 25% of the second cover audio file. Hosny et al. (2019) managed to distribute the secret message until around 90% of the cover audio. Next, Alsabhany et al. (2020) produced huge differences between certain parts of the difference signal produced by the usage of two or more different LSB levels for the embedding process of the secret message. In addition, there was also a small gap at around 70% of the signal difference which indicates this algorithm skipped this part of the embedding process. Meanwhile, BCM-LSB managed to distribute quite fairly without producing any huge differences between parts of the signal difference. The signal differences for the third cover audio file is shown in Figure 5.31.

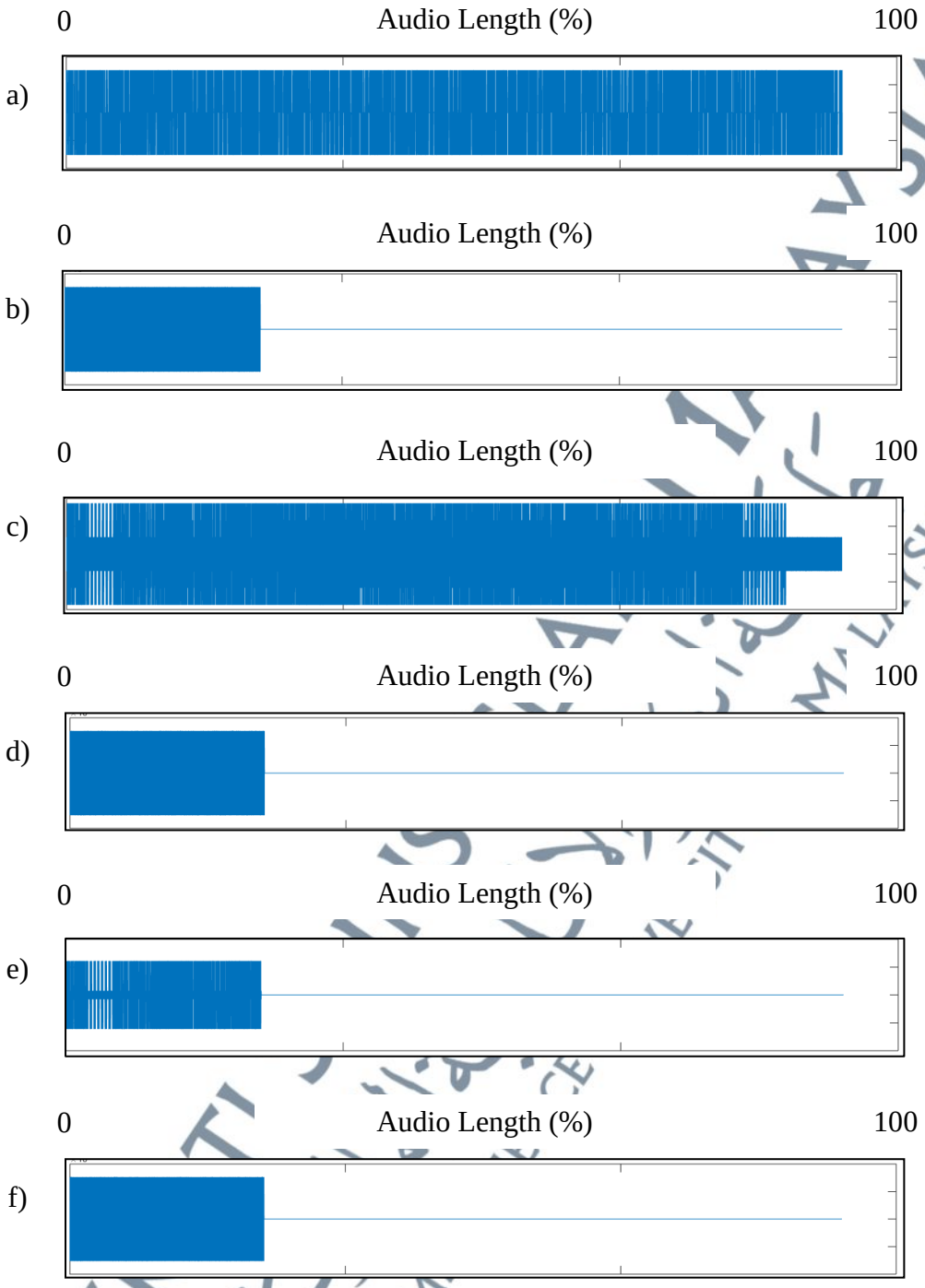


Figure 5.31: Difference Signals at 25% Capacity using Third Cover Audio File

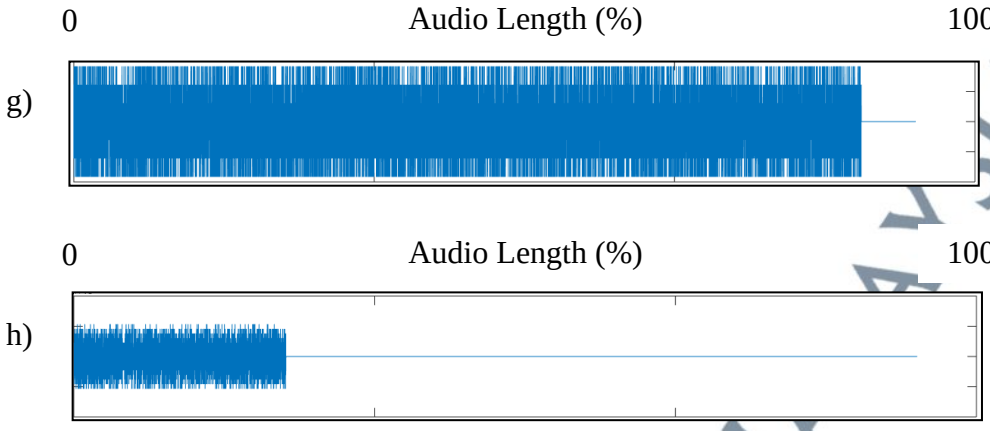


Figure 5.31: Difference Signals at 25% Capacity using Third Cover Audio File, continued

Based on Figure 5.31, except for all the similar trends discussed previously, it was worth noticing that only the BCM-LSB and Alsabhany et al. (2020) managed to distribute the secret message throughout the third cover audio file. However, Alhsabhany et al. (2020) produces differences at certain region indicating that there are different levels of embedding implemented during embedding process which is visible in this experiment.

In summary, BCM-LSB has major advantages in dynamic security compared to other algorithms. Despite some limitations in the dynamic performance of BCM-LSB when using different audio, it still outperformed sequential embedding behavior algorithms used in Indrayani (2020), Usanto (2022), Hashim et al. (2018), Bharti et al. (2019), and Xin et al. (2018). To improve the limitation, selecting a good cover audio is crucial as demonstrated by the results presented in Figure 5.27 to Figure 5.29.

5.3.1.5 Summary from Comparison Results of BCM-LSB Performance Against Existing LSB Embedding Algorithm

In general, BCM LSB produces high advantages against existing algorithms due to its flexibility changing its parameters such as *bps* and key *x*, *n* and *r*. However, it is

important to highlight the BCM-LSB's massive dynamic security performance against the existing algorithms due to its ability to distribute the secret message throughout the any cover used without depending much on the internal audio parameters such as audio sample's amplitude and the size of cover audio itself. Although there are fluctuations in the seg-SNR evaluations, it can be further improved by implementing cover audio selection algorithm.

5.3.2 Cover Selection Layer Comparison Results

This section is divided into two parts. Subsection 5.2.2.1 discusses the comparison of MCAS against existing cover selection algorithm on the dynamic security evaluation metric usage element while Subsection 5.2.2.2 discusses the comparison of MCAS against existing cover selection algorithm on the trade-off consideration element.

5.3.2.1 Dynamic Security Formulation Used for Cover Selection Evaluation Metric Comparison Results

The objective of this evaluation was to compare the usage of newly proposed dynamic security metric used in MCAS against existing dynamic security evaluation metric based on its performance and accuracy in terms of selecting superior solution. MCAS with similar imperceptibility and robustness evaluation metrics but different variations of dynamic security formulation used were compared in terms of the quantity and the quality of the solution found. The results for the cover selection used for the 4 KB secret message are shown in Table 5.14, Table 5.15 and Table 5.16.

Table 5.14: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ With Proposed Dynamic Security Formulation

Round	NPS	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	21	0	77.895	38.918
Round 2	21	0	77.895	38.918
Round 3	21	0	77.895	38.918

Table 5.15: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ With Existing Dynamic Security Formulation.

Round	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	10	0	67.0	33.0
Round 2	10	0	67.0	33.0
Round 3	10	0	67.0	33.0

Based on Table 5.14 and Table 5.15, MCAS with proposed dynamic security formulation and existing dynamic security formulation managed to achieve similar results of *NDS*, $Best_{Sol}$ and $Worst_{Sol}$ during each round but different *NPS*. Since not each set found the same solutions in their Pareto front, a new population was created by merging all these two Pareto fronts to find a new Pareto front using the newly proposed dynamic security. Table 5.16 shows the results of the new population created.

Table 5.16: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Formulation	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Proposed dynamic security evaluation metric in MCAS	21	0	77.908	39.250
Existing (Xin & Jiaojiao, 2018)	9	1	77.908	38.650

Based on Table 5.16, the newly proposed dynamic security produced better result which showed higher *NPS* lower *NDS* while obtaining similar result for $Best_{Sol}$. In term of NPS, it shows that newly proposed dynamic security evaluation metric managed to find more quality solution which do not dominated each other compared to using existing dynamic security evaluation proposed by Xin & Jiaojiao (2018) in MCAS. The current evaluation metric for dynamic security appears to be of lower quality as it indicates a greater number of NDS in comparison to the newly suggested metric. Moreover, the current metric identified a single dominant solution while the new metric did not. Both metrics yielded similar $Best_{Sol}$ values, hence, they demonstrated the most balanced and highest values in terms of the three characteristics. In conclusion, the newly proposed dynamic security metric utilized in MCAS was able to identify a greater number of non-dominated solutions, indicating a larger quantity of solutions with the same quality as the current metric. Next, the results for the cover selection used for the 8 KB secret message are shown in Table 5.17, Table 5.18 and Table 5.19.

Table 5.17: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ With Proposed Dynamic Security Formulation

Round	NPS	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	22	0	71.264	33.283
Round 2	22	0	71.264	33.283
Round 3	22	0	71.264	33.283

Table 5.18: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ With Existing Dynamic Security Formulation.

Round	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	8	0	67.0	33.0
Round 2	8	0	67.0	33.0
Round 3	8	0	67.0	33.0

Based on Table 5.17 and Table 5.18, MCAS with proposed dynamic security formulation and existing dynamic security formulation managed to achieve similar results of *NDS*, $Best_{Sol}$ and $Worst_{Sol}$ during each round. However, they produced different *NPS*. Since not each set find the same solutions in their Pareto front, a new population was created by merging all these two Pareto fronts to find a new Pareto front using the newly proposed dynamic security. Table 5.19 shows the results of the new population created.

Table 5.19: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Formulation	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Proposed dynamic security evaluation metric in MCAS	22	0	71.264	33.283
Existing (Xin & Jiaojiao, 2018)	8	0	67.687	34.035

Based on Table 5.19, the newly proposed dynamic security evaluation metric used in MCAS produced better result which showed higher *NPS* and $Best_{Sol}$ while producing similar result for *NDS*. In term of NPS, it shows that the newly proposed dynamic security evaluation metric managed to find more quality solution which are non-dominated of each other compared to using existing dynamic security evaluation used by Xin & Jiaojiao (2018) in MCAS. In term of $Best_{Sol}$, the proposed dynamic security

metric found solution which has higher performance and well-balanced in term of three characteristics. In summary, MCAS with the new dynamic security metrics was able to search and find higher number of non-dominated solutions which reflects a larger quantity of solutions with the same quality as the current metric. In addition, the solutions found also has higher quality which ranging from 33% to 71% compared to the existing method which ranging from 34% to 68%.

5.3.2.2 Trade-off Consideration During Cover Selection Comparison Results

The objective of this evaluation was to compare the performance and accuracy of each metrics in terms of selecting superior solution of MCAS algorithm (which consider the trade-off) against existing cover selection algorithm (which do not consider the trade-off) using 4 KB and 8 KB of secret message. To represent an algorithm that does not consider the trade-off, a new searching algorithm which implements brute-force technique searches and sorts each of the solutions based on characteristic mentioned previously in Chapter 4. Then, the number of solutions found by the MCAS was compared with the number of top solutions with different characteristics. The results for the cover selection used for the 4 KB secret message are shown in Table 5.20 and Table 5.21.

Table 5.20: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ of MCAS

Round	NPS	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	21	0	77.895	38.9181
Round 2	21	0	77.895	38.9181
Round 3	21	0	77.895	38.9181

Table 5.21: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Algorithm	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
MCAS	21	0	86.90	65.74
Brute force - imperceptibility Rashid, (2020)	6	15	76.59	63.13
Brute force - robustness Rashid, (2020)	2	19	84.51	48.56
Brute force - dynamic security Xin & Jiaojiao (2018)	3	18	66.94	29.98
Brute force – imperceptibility Wang et al. (2020)	0	21	85.43	61.11

Based on Table 5.21, the newly proposed dynamic security produced better result which showed higher *NPS* lower *NDS* and $Best_{Sol}$. In term of NPS, it shows that MCAS which consider the trade-off managed to find more quality solution which are non-dominated by each other compared to using existing cover selection algorithms which do not consider the trade-off. In addition to that, all existing cover selection algorithms also suggest low quality solution which is reflected by higher number of NDS compared to the MCAS as there is as high as 21 dominated solutions found by brute force – imperceptibility Wang et al. (2020). MCAS also found the highest $Best_{Sol}$, which shows that it found the highest in term of three characteristics. In summary, for 4KB of secret message, MCAS was able to identify a greater number of non-dominated solutions, indicating a larger quantity of solutions with the same quality as other existing algorithms. The results for the cover selection used for the 8 KB secret message are shown in Table 5.22 and Table 5.23.

Table 5.22: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ With Trade-off Consideration

Round	NPS	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	22	0	71.025	35.035
Round 2	22	0	71.025	35.035
Round 3	22	0	71.025	35.035

Table 5.23: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Algorithm	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
MCAS	22	0	87.74	65.70
Brute force - imperceptibility Rashid, (2020)	7	15	77.76	62.89
Brute force - robustness Rashid, (2020)	3	19	83.27	47.11
Brute force - dynamic security Xin & Jiaojiao (2018)	3	19	66.97	29.73
Brute force – imperceptibility Wang et al. (2020)	1	21	81.68	61.88

Based on Table 5.23, similar pattern is shown as the newly proposed dynamic security produced better result which showed higher *NPS* lower *NDS* and $Best_{Sol}$. In term of NPS, it shows that MCAS which consider the trade-off managed to find more quality solution which did not dominate each other compared to using existing cover selection algorithms which do not consider the trade-off. In addition to that, all existing cover selection algorithms also suggest low quality solution which reflect by higher number of NDS compared to the MCAS as there is as high as 21 dominated solutions found by brute force – imperceptibility Wang et al. (2020). MCAS also found the highest $Best_{Sol}$, which shows that it found the most balanced and highest in term of three characteristics. In summary, for 8 KB of secret message, MCAS was able to

identify a greater number of non-dominated solutions, indicating a larger quantity of solutions with the same quality as other existing algorithms.

In summary, for both 4 KB and 8 KB of secret message, MCAS provided better solutions compared to only considering one of the characteristics of audio steganography. The solutions found by MCAS found higher quality solutions which ranging from 66% to 87% compared to the highest existing method which only able found solutions with quality ranging from 47% to 83%. MCAS gave a well-rounded performance without neglecting other characteristics at any point hence consistently finding the highest normalized value solution which can produce better stego-file in all three characteristics.

5.3.3 Audio Steganography Model Comparison Results

This The CAS was compared with the BCM-LSB with manual cover selection audio steganography model. The objective of this evaluation is to prove that a cover selection algorithm is needed to enhance the audio steganography without depending on the user's knowledge. Since the user of audio steganography has various knowledge backgrounds on audio steganography, from zero knowledge up to professionally knowledgeable, the random search was implemented to mimic all user backgrounds. This evaluation was conducted to search cover selection based on 4 KB audio. The results for the cover selection used for the 4 KB secret message are shown in Table 5.24, Table 5.25 and Table 5.26.

Table 5.24: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using CAS

Round	NPS	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	26	70	77.6941182	33.8684211
Round 2	39	34	77.074994	38.979812
Round 3	33	24	77.721113	33.8449093

Based on Table 5.24 a newly proposed audio steganography which included MCAS produced a different result for each round. Round 3 found the highest the $Best_{Sol}$compared to two other rounds. Round 2 found the highest value of $Worst_{Sol}$. Next, round 1 produced the lowest NPS and the highest NDS. Each round did not manage to produce stable value because initial solutions in the first population were not the same. This resulted in the starting search areas for each round being different, hence providing different results. Next, all NPS solutions in each round were compiled to compared head-to-head with NPS solutions found by the random cover selection after this.

Next, the random selection was conducted in three rounds. NPS each round was combined, and another round of searching was conducted. The results for the combined random cover selection used for the 4 KB secret message are shown in Table 5.25.

Table 5.25: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ Using Random Cover Selection with BCM-LSB

Round	*NPS*	NDS	$Best_{Sol}$ (%)	$Worst_{Sol}$(%)
Round 1	14	36	83.3048	43.6301
Round 2	15	24	83.4590	32.9881
Round 3	17	23	83.2186	33.0000

Based on Table 5.25 this random cover selection produced different results for each round. Round 2 found the highest the $Best_{Sol}$compared to two other rounds. Round 1 found the highest value of $Worst_{Sol}$. Next, round 3 produced the highest NPS and the lowest NDS. Since all three rounds were random searching, it is understandable that each round did not manage to produce stable value. Next, all NPS solutions in each round were compiled to compared head-to-head with the solutions found by the CAS in Table 5.26.

Table 5.26: NPS, NDS, $Best_{Sol}$ and $Worst_{Sol}$ in New Population

Algorithm	***NPS***	**NDS**	$Best_{Sol}$ **(%)**	$Worst_{Sol}$**(%)**
CAS	34	64	54.695	28.345
Existing Audio Steganography Model	20	46	48.938	10.222

Based on Table 5.26, the newly proposed CAS algorithm produced a better result which showed higher *NPS*, higher $Worst_{Sol}$ and higher $Best_{Sol}$. Proposed CAS algorithm managed to prove that the cover selection is the best implementation instead of using manual selection which exposed to the decreasing any characteristic or worst, all characteristics. The solutions found by CAS found higher quality solutions which ranging from 28% to 54% compared to existing audio steganography model which only able found solutions with quality ranging from 10% to 48%. Hence, the key achievement is CAS algorithm managed to find $Best_{Sol}$, which is 5.757 percent higher than manual selection and the $Worst_{Sol}$ found by CAS is 18.123 percent compared to $Worst_{Sol}$ found by manual selection.

5.4 Chapter Summary

This research focuses on three aspects which are the LSB embedding level, cover selection level and audio steganography model level. The summary of all the results is presented in Table 5.27.

Table 5.27: Result Summary

Aspects	Experiment	Results
LSB embedding level	Imperceptibility characteristic performance comparison	BCM-LSB produces on par results with other existing methods that embed at lower level on SNR, MSE and PSR evaluation.
	Capacity characteristic performance comparison	BCM-LSB produces superior result on the maximum capacity comparison evaluation compared to other existing methods.
LSB embedding level	Robustness characteristic performance comparison	BCM-LSB produces superior result on against AWGN attack compared to the other existing methods
	Dynamic security characteristic performance comparison	BCM-LSB produces superior result on against seg-SNR spike test and difference signal test compared to the other existing methods
Cover Selection Layer	Dynamic security formulation used performance comparison	MCAS produces superior result compared to the existing method as it finds solutions with higher quality that ranging from 33% to 71% compared to the existing method's quality that ranging from 34% to 68%.
	Trade-off considerations during cover selection performance comparison	MCAS produces superior result compared to the existing method as it finds solutions with higher quality solutions that ranging from 66% to 87% compared to the highest existing method which only able found solutions with quality that ranging from 47% to 83%.

Aspects	Experiment	Results
Audio Steganography Model	Audio steganography model performance comparison evaluation	CAS produces superior result compared to the existing model as it finds solutions with higher quality solutions which ranging from 28% to 54% compared to existing audio steganography model which only able found solutions with quality ranging from 10% to 48%.

In general, this research specifically focused on the dynamic security issue and performed two types of tests to compare the dynamic security of the proposed BCM-LSB. It surpassed the existing works in terms of dynamic security by maintaining the spreading of the secret message across the cover audio compared to the existing research. For cover selection level, the proposed MCAS surpassed all the existing cover selection algorithms by considering the trade-off among all the measured characteristics and introducing new dynamic security evaluation metric. Lastly, for audio steganography model level, the proposed CAS prove that the state-of-art audio steganography model needed cover audio selection algorithm in order to improve the characteristics of stego-file produced as the user who does not has background about audio steganography tend to choose low quality cover for audio steganography.