



Metaheuristics with Vector Quantization Enabled Codebook Compression Model for Secure Industrial Embedded Environment

Adepu Shravan Kumar and S. Srinivasan*

Department of Biomedical Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Saveetha Nagar, Thandalam, Chennai, 602105, India

*Corresponding Author: S. Srinivasan. Email: srinivasans.sse@saveetha.com

Received: 08 October 2022; Accepted: 06 December 2022

Abstract: At the present time, the Industrial Internet of Things (IIoT) has swiftly evolved and emerged, and picture data that is collected by terminal devices or IIoT nodes are tied to the user's private data. The use of image sensors as an automation tool for the IIoT is increasingly becoming more common. Due to the fact that this organisation transfers an enormous number of photographs at any one time, one of the most significant issues that it has is reducing the total quantity of data that is sent and, as a result, the available bandwidth, without compromising the image quality. Image compression in the sensor, on the other hand, expedites the transfer of data while simultaneously reducing bandwidth use. The traditional method of protecting sensitive data is rendered less effective in an environment dominated by IIoT owing to the involvement of third parties. The image encryption model provides a safe and adaptable method to protect the confidentiality of picture transformation and storage inside an IIoT system. This helps to ensure that image datasets are kept safe. The Linde–Buzo–Gray (LBG) methodology is an example of a vector quantization algorithm that is extensively used and a relatively new form of picture reduction known as vector quantization (VQ). As a result, the purpose of this research is to create an artificial humming bird optimization approach that combines LBG-enabled codebook creation and encryption (AHBO-LBGCCE) for use in an IIoT setting. In the beginning, the AHBO-LBGCCE method used the LBG model in conjunction with the AHBO algorithm in order to construct the VQ. The Burrows-Wheeler Transform (BWT) model is used in order to accomplish codebook compression. In addition, the Blowfish algorithm is used in order to carry out the encryption procedure so that security may be attained. A comprehensive experimental investigation is carried out in order to verify the effectiveness of the proposed algorithm in comparison to other algorithms. The experimental values ensure that the suggested approach and the outcomes are examined in a variety of different perspectives in order to further enhance them.

Keywords: Codebook compression; industrial internet of things; lbg model; metaheuristics; vector quantization



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

The Industrial Internet of Things (IIoT) is the primary impetus behind the development of the 6th generation (6G) mobile communications infrastructure [1]. A new paradigm will be presented by an artificial intelligence (AI) supported 6G network, which will promote the inclusion of vertical industry in various scenarios such as industrial digital transformation and smart manufacturing [2]. The evolutionary 6G network needs to offer ultra-high data rate with ultra-low latency, reliability, ubiquitous intelligence, and massive connections [3], which is supporting the intelligent production scheduling, prediction maintenance of equipment, high-precision product quality control, and other value-added services in a variety of industrial IoT systems, which is giving rise to the fourth industrial revolution [4]. People are becoming more concerned about the security of their data in light of the phenomenal rise of the Internet. Several other strategies were presented as potential means of protecting sensitive information. Another method of data protection is known as data concealing [5]. In order to achieve the goal of information security, data hiding involves embedding large datasets within carriers. This prevents the information from being recognised and so helps to achieve the goal [6].

The technique of data concealing may be divided into two categories: irreversible and reversible, and it is often used in cloud applications for the purpose of accessorial data embedding, namely the embedding of keywords for the purpose of picture retrieval [7]. As a result, one of the key concerns is to maximise payload while minimising distortion. The vector quantization (VQ) system is a lossy image compression method that may be broken down into three stages: image compression, decompression, and codebook training. Image compression is the first step. The LINDO-BUZO-GRAY (LBG) technique has been used extensively in the creation of conventional codebooks [8]. It has been proposed that research workers should speed up the index code search as well as the design of the codebook. After using the codebook as a guide, the picture is compressed into an index table. During the decompression process, the index table is converted into an image by using codebooks that are quite similar to one another [9]. The VQ data source coding technology is efficient and reliable. The LBG approach was named after the individuals who first proposed it, and there are also a number of additional improved variants [10]. This technology can provide a straightforward decoding strategy in addition to a greater compression ratio.

In this work, an artificial humming bird optimization with LBG enabled codebook creation and encryption approach known as AHO-LBGCCE is developed for use in an IIoT setting. In the beginning, the AHBO-LBGCCE method used the LBG model in conjunction with the AHBO algorithm in order to construct the VQ. The Burrows-Wheeler Transform (BWT) model is used in order to accomplish codebook compression. In addition, the Blowfish algorithm is used in order to carry out the encryption procedure so that security may be attained. A comprehensive experimental investigation is carried out in order to verify the effectiveness of the proposed algorithm in comparison to other algorithms.

2 Related Works

When [11] is being analysed, the WOA may be used to determine which image compression yields the superior codebook. WOA is a standard method that takes into account a variety of searching methods and is used to construct a robust codebook from picture compressions. The capability of the proposed system to compress a variety of typical pictures reveals that the proposed approach is capable of producing image compressions with the appropriate characteristics. The approaches such as PSO, BA, and FA that were described were superseded by the superior compression capabilities of the system that was demonstrated. An MRFO employing an LBG-related microarray image compression (MRFOLBG-MIC) technique is investigated by Alkhaldi et al. [12]. In most cases, the LBG technique was used in order to construct a local optimal codebook for image compressions. The MRFO method, which is designed to produce better codebooks, was used in order to improve the LBG-VQ codebook that was developed.

An effective method for compressing biomedical images using a CC platform and HHO-based LBG techniques was proposed by Satish Kumar et al. [13]. The method that is provided allows for a seamless transition between the exploitation phase and the exploration phase. A delicate method for masking digital picture data that is reliant upon lattice VQ (LVQ) was provided in reference [14]. The picture was converted from the spatial domain to the frequency domain using an integer-to-integer discrete wavelet transform (DWT) during the data concealing system that was provided, as well as during a primary preprocessing phase (IIDWT). After that, the LVQ lattices of A4 and Z4 were applied to embed data into the cover picture. The authors of [15] are interested in steganography using Internet Low Bitrate Codec voice streams as their primary emphasis. The sensitive information was encoded together with an improvement to the searching range of gain codebooks, which was done in order to quantize the gain values.

A hardware-effective image compression circuit that is reliant on VQ to maximum-speed image sensors is investigated in Huang et al.'s [16] research. During the course of this circuit, a SOM was carried out to facilitate the on-chip learning of codebooks in order to flexibly meet the requirements of many applications. The authors propose a reconfigurable complete-binary adder tree as a means of decreasing the amount of hardware resources required, during which time the arithmetic unit was employed in its entirety. Establishing a new hierarchical geometry approach that is reliant on adaptable Tree-Structured Point-LVQ is the work of Filali et al. [17] (TSPLVQ). This technology enables the structuring of three-dimensional content in a hierarchical fashion, which improves the compression efficiency of static point clouds. Two different techniques of quantization were used in order to show recursively the three-dimensional point clouds as a succession of embedding truncated cubic lattices. This was done in accordance with the adaptive and multiscale architecture that the system had.

3 The Proposed Model

In this piece, we present a new AHBO-LBGCCE system that we have developed for the construction and compression of secure codebooks in the context of the IIoT environment. In the beginning, the AHBO-LBGCCE method used the LBG model in conjunction with the AHBO algorithm in order to construct the VQ. In this particular study, the BWT model is utilised for the purpose of codebook compression. In addition, the Blowfish algorithm is utilised as part of the encryption process in order to accomplish the goal of achieving security.

3.1 AHBO Based Codebook Construction Process

VQ can be determined by the block coding mechanism which is used to compress images with loss of data. In the presented method, the construction of codebooks is a crucial process [18]. Consider $Y = \{y_{ij}\}$ as the size of the raw images of $M \times M$ pixel that are subdivided into distinct block sizes of $n \times n$ pixel. The input vector can be characterized as row vectors $x_i = (x_{i1}, x_{i2}, \dots, x_{iL})$ and j -th codewords of the codebooks can be defined as $c_j = (c_{j1}, c_{j2}, \dots, c_{jL})$. Enhancing C through MSE established the minimization of D deformation function. The minimal value of D indicates better C.

$$D(C) = \frac{1}{N_b} \sum_{j=1}^{N_c} \sum_{i=1}^{N_b} \mu_{ij} \cdot \|x_i - c_j\|^2 \quad (1)$$

Subjected to the constraint provided by the subsequent equation:

$$\sum_{j=1}^{N_c} \mu_{ij} = 1, \quad \forall i \in \{1, 2, \dots, N_b\} \quad (2)$$

$$\mu_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is in the } j\text{th cluster} \\ 0 & \end{cases} \quad (3)$$

$L_k \leq C_{jk} \leq U_k$, $k = 1, 2, \dots, L$, in which L_k and U_k symbolizes the lower and upper k^{th} components in input vector. The $\|x_i - c_i\|$ displays Euclidean distance between x vector and C codeword. Fig. 1 represents the overall VQ process.

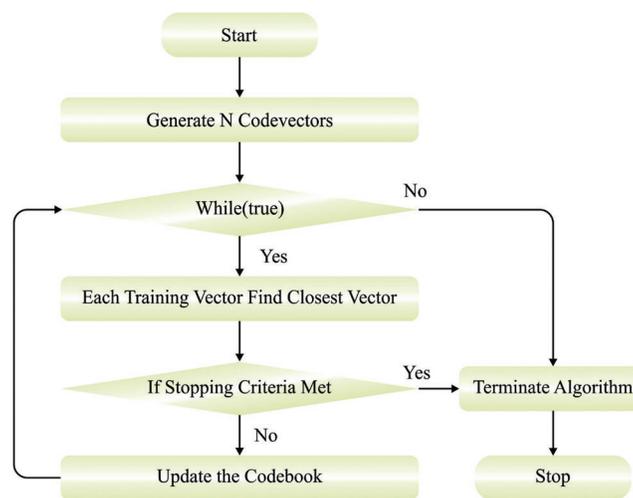


Figure 1: Overall VQ process

3.1.1 LBG Algorithm

In 1957, Lloyd developed an LBG algorithm that can be characterised as an approach to scalar quantization. In 1980, this algorithm was generalised to be used with VQ. In order to define the codebook, it applies two conditions that have been predetermined to the input vector. The LBG approach uses two conditions in an iterative fashion in order to arrive at the optimal codebook based on the particular procedure.:

- Split the input vectors into various groups via minimal distance rules. The resulting blocks are stored in binary indicator matrixes whereby the component is formulated by Eq. (4),

$$\mu_{ij} = \begin{cases} 1 & \text{if } d(x_i, c_j(k)) = \min_p d(x_i, c_p(k)) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

- Find the centroid of each portion. The former codeword was replaced with the presented centroid:

$$c_j(k+1) = \frac{\sum_{i=1}^{N_b} \mu_{ij} \cdot x_i}{\sum_{i=1}^{N_b} \mu_{ij}} \quad (5)$$

- Go to step 1 until no modification in c_j takes place.

3.1.2 AHBO Based LBG Process

The AHBO is a biologically stimulated optimization approach that is based on the actions of intelligent hummingbirds (HB) [19]. [Citation needed] In the following, the three AHBO tactics that are considered to be the most essential will be discussed.

Sources of food: It performs frequent evaluations of the properties of the food sources in order to select an appropriate source from among a group of food sources. These evaluations include factors such as the amount of time that has passed since the bees last visited the flowers, the rate at which the nectar refills, the composition of each individual flower, and the quality of the nectar.

Hummingbird: If each HB is assigned to specific food sources where it may be fed, then the HB and the food sources will be in a position that is comparable to one another. A HB is able to recall the pace at which various food sources and places refresh their nectar supply, and they may relay this knowledge to other members of the colony. In addition, HB is able to recall how much longer it has been since they have visited the food sources on their own.

Visit table: This information was recorded as the visit level to the food source for HB, which represents the number of times that have passed since the HB's most recent visit to the food source. That particular HB would give more emphasis to its visits to food sources that get a big volume of visitors.

The objective of the AHBO algorithm is to achieve victory over optimal problems using a swarm-based metaheuristic method. This section presents three mathematical models that represent three different foraging techniques used by HB. These methods include territorial foraging, migratory foraging, and directed foraging. The suggested method, much like the optimizer in the swarm-based class, may be broken down into three primary stages. The behaviour of HB in both flying and foraging served as inspiration for the optimization strategy that is being given here. The AHBO algorithm's three foundational phases, which are currently under discussion, are as follows: It is stated in the following equation how directed foraging employs three different combat techniques in foraging: diagonal flight, axial flight, and omni-directional flight.:

$$v_i(t+1) = x_{i,ta}(t) + h.b.(x_i(t) - X_{i,ta}(t))h \sim N(0, 1) \quad (6)$$

In Eq. (2), $x_{i,ta}(t)$ epitomizes the location of target food source, h indicates the guiding factor, and $x(t)$ denotes the location of i -th food sources at time t and it can be expressed in the following:

$$x_{Ai}(t) = \begin{cases} x_i(t) f(x_i(t)) \leq f(v_i(t+1)) \\ v_i(t+1) f(\chi_{i(t)}) > f(v_i(t+1)) \end{cases} \quad (7)$$

While $f(x(t))$ and $f(v_i(t+1))$ correspondingly represents the function fitness value for $x(t)$ and $v_i(t+1)$. Fig. 2 depicts the steps involved in AHBO technique.

Territorial foraging: The succeeding expression indicates the local searching of HB:

$$v_i(t+1) = \chi_{i(t)} + g.b.(\chi_i(t))g \sim N(0, 1) \quad (8)$$

In Eq. (8), g signifies the territorial factor. The migration foraging of HBs can be mathematically formulated as follows:

$$x_{wor}(t+1) = lb + r.(ub - lb) \quad (9)$$

In Eq. (9), x_{wor} means the food source with worst population rate of nectar refilling, r implies a random factor, ub and lb correspondingly indicate the upper and lower boundaries.

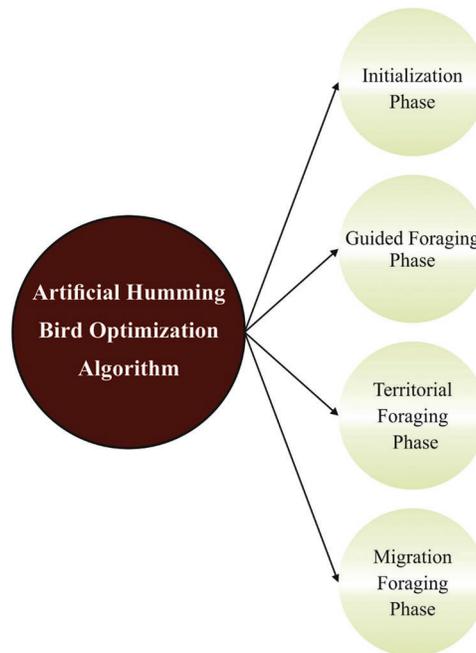


Figure 2: Steps involved in AHBO algorithm

There has been no evidence shown to show how the AHBO-based LBG models really operate. It is necessary to first cut the input picture into non-overlapping blocks before using the LBG technique to the quantization process [20]. The codebook was implemented with the help of the LBG model, which has the capability of being trained using the AHBO technique and has to be compliant with the restrictions of global convergence. The index number is sent through the communications media, and then it is reconstructed at the destination using the decoder. The accompanying codewords and the reformed index have been correctly placed in order to generate a decompressed picture with a size that is substantially comparable to the input image that was supplied. The AHBO model has been discussed in the following steps.

Step1: Initializing parameters: now, the codebook created through LBG approach is declared the primary solution where remaining solution is randomly developed. The solution attained denotes a codebook of N_c codewords.

Step2: (Selective current optimum solution): The fitness of every solution can be processed according to Eq. (1) and chooses the high fitness position as better outcome.

Step3: (Generation of novel solution): The grey wolf location is upgraded. Once the (K) arbitrarily created number is greater than $\sim a$ the bad location can be replaced with the recently discovered location and keep the optimum position unchanged.

Step4: Rank the solution under the application of fitness functions and choose the best solution.

Step5: (Ending condition): Follow steps 2 and 3 until the Stopping condition is reached.

3.2 Codebook Compression Process

In the presented AHBO-LBGCCE technique, the compression of codebook process is carried out using the BWT technique. Let us $\Sigma = \{A, C, G, T\}$ is the alphabet that generates the order and is a symbol lesser when compared to the lexicographic sequence of every character in it [21]. To provide a string $S = a_0a_1, \dots, a_n$, whereas $a_n =$, such that $S[i] = a_i$ defines the i^{th} letter, $S[i, j] = a_i, \dots, a_j$ is a sub-sequence

of s , and $S_i = S[i, n - 1]$ is a suffix of S . S refers to the BWT infrastructure that is outcome of n -step right shift function on S , a matrix array of n rows, and one character at a time is attained. All the rows from the array are a resultant of S -rightshift function, and the string beforehand $\$$ character is the suffixes equivalent to row. Afterward, the matrix was introduced, and the suffix of all the rows from the matrix was arranged based on the dictionary order, later rearranged for obtaining the last conversion array. This character in the final column of matrix is integrated for obtaining the converting string order. The subsequent computation of $BWT(S)$ to offer string S contains 3 fundamental steps:

Step1: add a $\$$ symbol to end of S that is lesser than S symbol.

Step2: Generate the M matrixes, the row1 of that is equivalent to S . Row2 of M matrix has shifted 1bit cyclically towards right of row1 of M matrix (S value). Row3 of M matrix has shifted 1bit cyclically towards right of row2 of M matrix, and it can be repeated still attain row n . Through the observe of matrixes, it could be initiated that once the n^{th} row of matrix shift to right by 1bit, the 1st row S is attained again that represents the structure cyclical shift has been implemented still the n^{th} row, concluding a rotation of sequence S .

Step3: Create the converting text $S=BWT(S)$ by adopting the final column of M matrix.

3.3 Encryption Using Blowfish

The Blowfish (BF) method is used here as part of the process of encrypting data. BF is a blockcipher with a symmetric key that has a key length that may vary from 32–448 bits and a block size that can be up to 64 bits [22]. The BF encryption system was completely open-source, so anybody could use it without paying a fee, and it wasn't protected by any patents. The encrypting software known as SplashID was created by combining BF with numerous other tools. The BF security approach went through rigorous testing. In addition, the BF is one of the quickest components built in order for everyone to use it. It is capable of doing important tasks, such as SplashID, which are compatible with a number of different processor types. While the symmetric blockcipher known as BF is used as a casual alternative to DES or IDEA, these algorithms are still widely used. It makes use of a variable-length key with a length somewhere between 32 and 448 bits, making it suitable for both domestic and commercial applications. BF is a quick and cost-free alternative to the most common encrypting methods. Bruce Schneier was the one that developed it. As a result, it is capable of being extensively analysed, and since it is a reliable encryption system, it has the potential to gradually gain popularity. The BF is not protected by a patent, has an open licence, and does not restrict its users in any way. Because there are not many primitives that are capable of operating on devices with little CPU capacity, lightweight cryptography has become more significant in recent years.

4 Results and Discussion

In this section, the experimental validation of the AHBO-LBGCCE technique is carried out on a set of images. Table 1 offers a comprehensive PSNR examination of the AHBO-LBGCCE model under varying levels of bit rate (BR).

Fig. 3 represents a brief PSNR examination of the AHBO-LBGCCE model with diverse BR levels under image 1. The results inferred that the AHBO-LBGCCE method has reached increased PSNR values under each BR. For example, with BR of 0.1875, the AHBO-LBGCCE technique has gained maximum PSNR of 28.01 dB. In contrast, the hybrid LZMA, CSA Model_LBG, FFA Model_LBG, HBMO Model_LBG, and QPSO Model_LBG approaches have obtained reduced PSNR values of 27.02, 25.68, 24.31, 23.06, and 21.84 dB correspondingly.

Table 1: PSNR analysis of AHBO-LBGCCE system with distinct images and BR

No. of images	Methods	PSNR (dB)							
		Bit rate							
		0.1875	0.25	0.3125	0.375	0.4375	0.5	0.5625	0.625
Image1	AHBO-LBGCCE	28.01	29.67	31.43	32.50	33.69	34.68	36.41	37.70
	Hybrid LZMA Model	27.02	28.12	29.71	31.66	32.08	33.62	35.53	36.33
	CSA model-LBG	25.68	26.63	28.28	29.95	30.89	32.35	33.97	34.73
	FFA model-LBG	24.31	24.85	26.79	28.20	29.41	31.50	32.67	33.57
	HBMO model-LBG	23.06	23.75	25.66	27.22	27.84	30.58	31.66	32.77
	QPSO model-LBG	21.84	22.68	23.86	25.47	26.14	28.83	30.86	31.29
Image2	AHBO-LBGCCE	25.89	26.88	28.28	29.50	30.80	31.72	33.50	35.15
	Hybrid LZMA model	24.21	25.55	26.97	28.70	29.72	30.15	31.94	33.99
	CSA model-LBG	22.92	24.04	25.70	27.55	28.63	28.80	31.12	33.01
	FFA model-LBG	21.77	22.45	24.22	26.42	27.12	27.04	29.44	31.68
	HBMO model-LBG	20.34	20.65	22.83	24.84	26.14	25.93	28.51	30.31
	QPSO model-LBG	19.37	18.97	21.77	23.97	24.98	24.64	27.42	28.95
Image3	AHBO-LBGCCE	27.70	29.03	30.23	31.21	32.03	32.99	34.10	34.92
	Hybrid LZMA model	25.98	27.91	28.44	30.32	30.48	31.47	32.41	34.09
	CSA model-LBG	25.04	26.38	26.96	28.55	29.60	30.16	30.70	32.72
	FFA model-LBG	23.80	24.94	25.70	27.54	28.46	29.08	29.29	31.21
	HBMO model-LBG	22.64	23.52	24.02	26.45	26.72	27.40	28.40	30.12
	QPSO model-LBG	21.65	22.40	22.50	24.68	25.26	26.59	27.52	28.32
Image4	AHBO-LBGCCE	27.95	29.63	31.39	32.32	33.66	34.73	36.04	37.55
	Hybrid LZMA model	26.87	28.21	29.79	31.13	32.36	33.83	34.67	36.37
	CSA model-LBG	26.01	26.86	28.33	30.24	31.55	32.16	33.73	35.11
	FFA model-LBG	24.74	25.34	27.43	29.22	29.99	31.22	32.03	33.40
	HBMO model-LBG	23.93	23.96	26.05	27.90	28.25	29.60	31.22	32.60
	QPSO model-LBG	23.00	22.65	25.13	26.91	27.31	28.64	30.41	31.30
Image5	AHBO-LBGCCE	27.73	28.65	29.11	30.97	31.59	32.18	33.11	35.43
	Hybrid LZMA model	24.64	25.33	26.44	27.40	29.03	31.01	31.67	33.14
	CSA model-LBG	23.02	24.06	24.89	25.72	27.30	29.61	30.87	31.78
	FFA model-LBG	21.60	22.44	23.76	24.34	25.80	28.16	29.67	30.03
	HBMO model-LBG	19.98	21.52	22.89	22.83	24.67	27.16	28.14	28.43
	QPSO model-LBG	18.49	20.49	22.05	21.13	23.03	25.92	26.45	27.01

Fig. 4 exhibits a detailed PSNR inspection of the AHBO-LBGCCE approach with varied BR levels under image 2. The outcome demonstrates that the AHBO-LBGCCE technique has obtained maximum PSNR values under each BR. For example, with BR of 0.1875, the AHBO-LBGCCE approach has

resulted in maximum PSNR of 25.89 dB. On the other hand, the hybrid LZMA, CSA Model_LBG, FFA Model_LBG, HBMO Model_LBG, and QPSO Model_LBG techniques have gained least PSNR values of 24.21, 22.92, 21.77, 20.34, and 19.37 dB correspondingly.

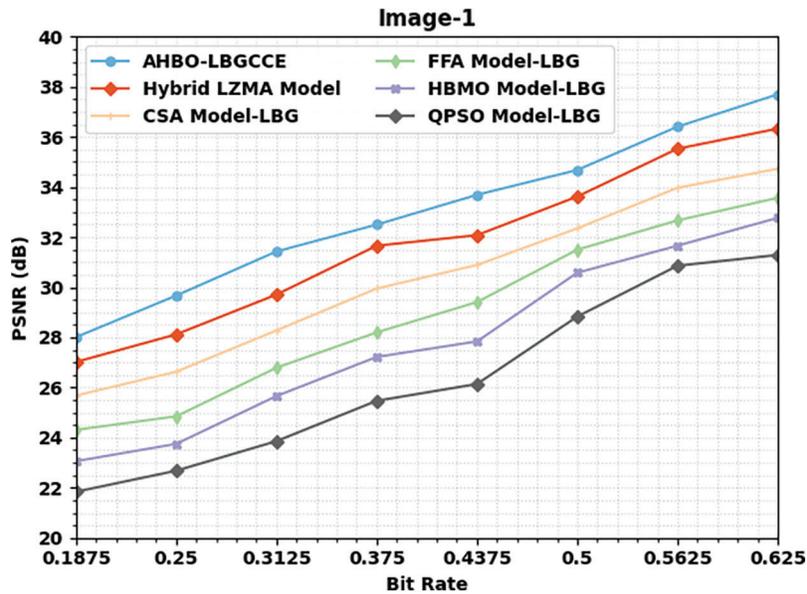


Figure 3: PSNR analysis of AHBO-LBGCCE system under image 1

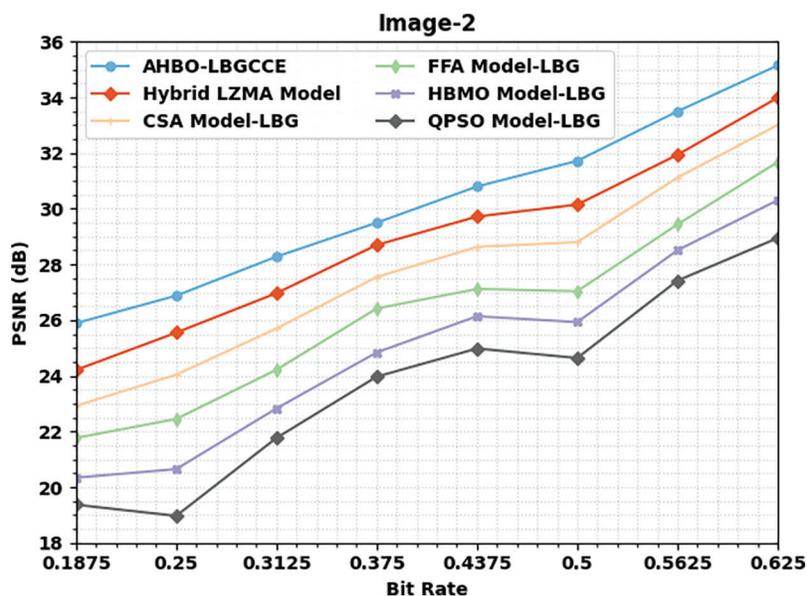


Figure 4: PSNR analysis of AHBO-LBGCCE system under images

Fig. 5 illustrates a comparative PSNR study of the AHBO-LBGCCE technique with several BR levels under image 3. The outcomes described that the AHBO-LBGCCE approach has resulted in raised PSNR values in each BR. For example, with BR of 0.1875, the AHBO-LBGCCE approach has provided increased PSNR of 27.70 dB. Contrastingly, the hybrid LZMA, CSA Model_LBG, FFA Model_LBG,

HBMO Model_LBG, and QPSO Model_LBG techniques have attained decreased PSNR values of 25.98, 25.04, 23.80, 22.64, and 21.65 dB correspondingly.

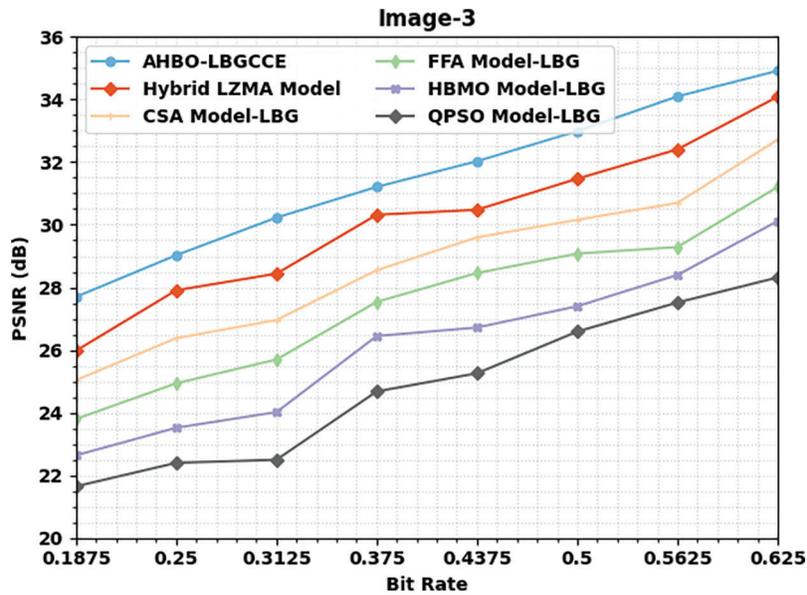


Figure 5: PSNR analysis of AHBO-LBGCCE system under image3

Fig. 6 studies comparative PSNR outcomes of the AHBO-LBGCCE model with different BR levels under image 4. The experimental values denoted that the AHBO-LBGCCE model has resulted in better PSNR values under each BR. For example, with BR of 0.1875, the AHBO-LBGCCE technique has accomplished highest PSNR of 27.95 dB. Simultaneously, the hybrid LZMA, CSA Model_LBG, FFA Model_LBG, HBMO Model_LBG, and QPSO Model_LBG techniques have demonstrated reduced PSNR values of 26.87, 26.01, 24.74, 23.93, and 23.00 dB correspondingly.

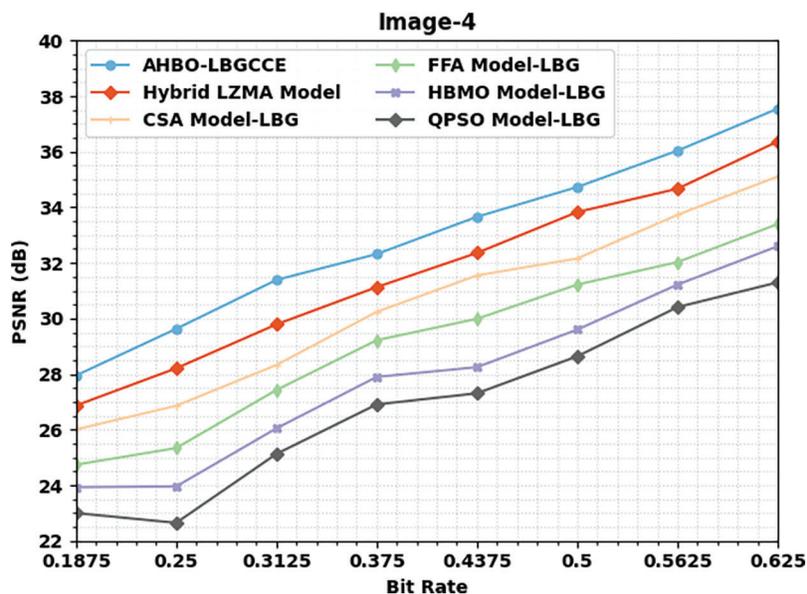


Figure 6: PSNR analysis of AHBO-LBGCCE system under image4

In Fig. 7, a comprehensive PSNR analysis of the AHBO-LBGCCE method with several BR levels under image 5. The experimental outcomes represented that the AHBO-LBGCCE technique has attained raised PSNR values in each BR. For example, with BR of 0.1875, the AHBO-LBGCCE approach has demonstrated greater PSNR of 27.73 dB. In contrast, the hybrid LZMA, CSA Model_LBG, FFA Model_LBG, HBMO Model_LBG, and QPSO Model_LBG techniques have exhibited inferior PSNR values of 24.64, 23.02, 21.60, 19.98, and 18.49 dB correspondingly.

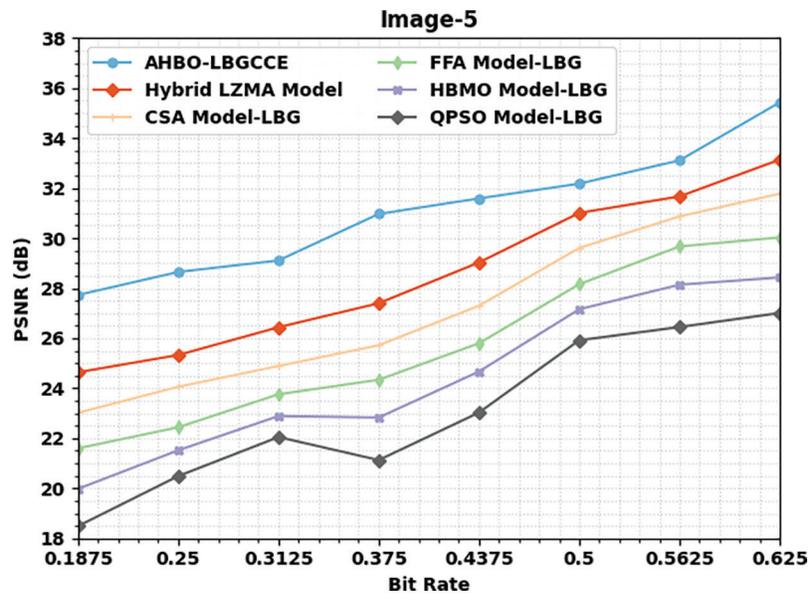


Figure 7: PSNR analysis of AHBO-LBGCCE system under image5

Next, an average PSNR study of the AHBO-LBGCCE model is performed in Table 2 and Fig. 8. The outcomedemonstrates the improved outcomes of the AHBO-LBGCCE model with maximum values of PSNR under all test images. It is noticeable the FFA Model_LBG, HBMO Model_LBG, and QPSO Model_LBG has demonstrated least PSNR values. Though the hybrid LZMA and CSA Model_LBG stated nearer average PSNR values, the AHBO-LBGCCE model has outpaced the other ones.

Table 2: Average PSNR analysis of AHBO-LBGCCE system with other recent methodologies in distinct images

Average PSNR (dB)					
Methods	Image_1	Image_2	Image_3	Image_4	Image_5
AHBO-LBGCCE	33.01	30.22	31.53	32.91	31.10
Hybrid LZMA model	31.76	28.90	30.14	31.65	28.58
CSA model-LBG	30.31	27.72	28.76	30.50	27.16
FFA model-LBG	28.91	26.27	27.50	29.17	25.73
HBMO model-LBG	27.82	24.94	26.16	27.94	24.45
QPSO model-LBG	26.37	23.76	24.87	26.92	23.07

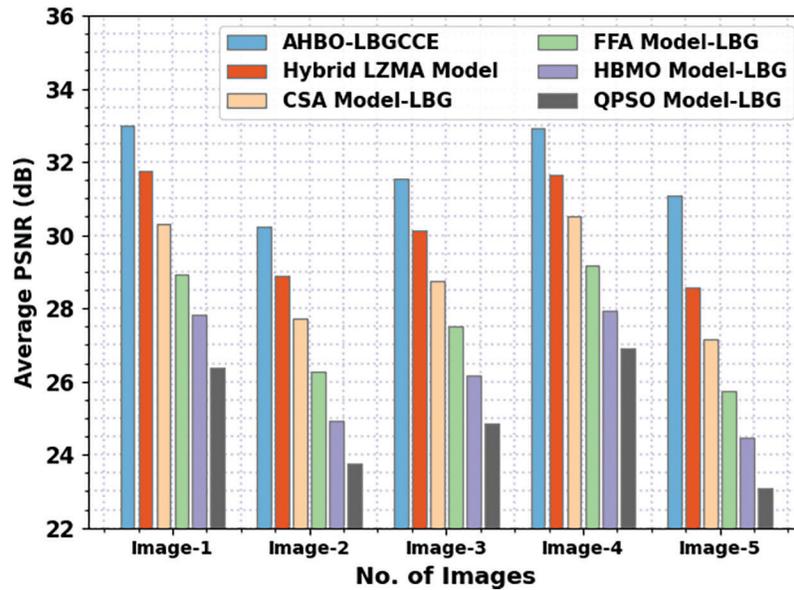


Figure 8: Average PSNR analysis of AHBO-LBGCCE system with distinct images

Finally, an average computation time (ACT) examination of the AHBO-LBGCCE system is studied in Table 3 and Fig. 9.

Table 3: ACT analysis of AHBO-LBGCCE system with other recent methodologies in distinct images

Methods	Average computation time (sec)				
	Image_1	Image_2	Image_3	Image_4	Image_5
AHBO-LBGCCE	0.257	0.260	0.254	0.263	0.277
Hybrid LZMA Model	0.272	0.273	0.272	0.276	0.290
CSA Model-LBG	0.287	0.295	0.284	0.297	0.313
FFA Model-LBG	0.303	0.310	0.302	0.311	0.325
HBMO Model-LBG	0.315	0.331	0.322	0.329	0.340
QPSO Model-LBG	0.332	0.344	0.340	0.342	0.362

The result indicates that the AHBO-LBGCCE approach has obtained superior performance with minimal values of ACT under all test images. It is also noted that the FFA Model_LBG, HBMO Model_LBG, and QPSO Model_LBG have shown poor performance. Although the hybrid LZMA and CSA Model_LBG reported closer ACT values, the AHBO-LBGCCE model has outperformed the other ones. These results assured the enhanced performance of the AHBO-LBGCCE method over other recent approaches.

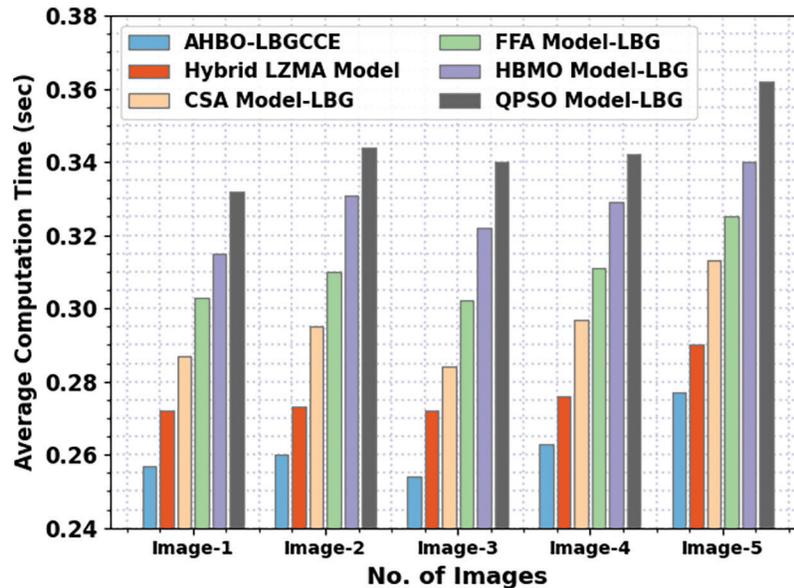


Figure 9: ACT analysis of AHBO-LBGCCE system under distinct images

5 Conclusion

In the course of this research, we came up with an innovative AHBO-LBGCCE method for the secure generation and compression of codebooks used in the context of the IIoT. In the beginning, the AHBO-LBGCCE method used the LBG model in conjunction with the AHBO algorithm in order to construct the VQ. In this particular research, the BWT model is used for the purpose of codebook compression. In addition, the Blowfish algorithm is used as part of the encryption process in order to accomplish the goal of achieving security. A comprehensive experimental investigation is carried out in order to verify the effectiveness of the provided technique in comparison to other algorithms. The experimental data helped to verify that the suggested approach might be improved, and the findings are being evaluated under a variety of different criteria. Using hybrid metaheuristics will allow for the performance of the AHBO-LBGCCE approach to be improved in the future.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. Löhdefink, A. Bär, J. Sitzmann and T. Fingscheidt, "Adaptive bitrate quantization scheme without codebook for learned image compression," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, pp. 1732–1737, 2022.
- [2] X. Zhang, F. An, L. Chen, I. Ishii and H. J. Mattausch, "A modular and reconfigurable pipeline architecture for learning vector quantization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3312–3325, 2018.
- [3] U. Jayasankar, V. Thirumal and D. Ponnurangam, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 2, pp. 119–140, 2021.
- [4] N. Floropoulos and A. Tefas, "Complete vector quantization of feedforward neural networks," *Neurocomputing*, vol. 367, no. 1, pp. 55–63, 2019.

- [5] J. Ravishankar, M. Sharma and S. Khaidem, "A novel compression scheme based on hybrid tucker-vector quantization via tensor sketching for dynamic light fields acquired through coded aperture camera," in *Proc. of the Int. Conf. on 3D Immersion (IC3D)*, Brussels, Belgium, IEEE, pp. 1–8, 2021.
- [6] J. Vizárraga, R. Casas, A. Marc and J. D. Buldain, "Dimensionality reduction for smart IoT sensors," *Electronics*, vol. 9, no. 12, pp. 2035, 2020.
- [7] S. Nousias, E. V. Pikoulis, C. Mavrokefalidis and A. S. Lalos, "Accelerating deep neural networks for efficient scene understanding in automotive cyber-physical systems," in *Proc. of the 4th IEEE Int. Conf. on Industrial Cyber-Physical Systems (ICPS)*, Victoria, BC, Canada, IEEE, pp. 63–69, 2021.
- [8] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis *et al.*, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 1–26, 2021.
- [9] S. M. Darwish and A. A. Almajtomi, "Metaheuristic-based vector quantization approach: A new paradigm for neural network-based video compression," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7367–7396, 2021.
- [10] Y. Jeon, A. M. Mohammadi and L. Namyoon, "Communication-efficient federated learning over MIMO multiple access channels," *IEEE Transactions on Communications*, vol. 70, no. 10, pp. 6547–6562, 2022.
- [11] J. Rahebi, "Vector quantization using whale optimization algorithm for digital image compression," *Multimedia Tools and Applications*, vol. 2022, pp. 1–27, 2022.
- [12] N. A. Alkhalidi, R. A. A. Alsedais, H. T. Halawani and S. M. A. Aboutaleb, "Manta ray foraging optimization with vector quantization based microarray image compression technique," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–18, 2022.
- [13] T. Satish Kumar, S. Jothilakshmi, B. C. James, M. Prakash, N. Arulkumar *et al.*, "HHO-based vector quantization technique for biomedical image compression in cloud computing, International," *Journal of Image and Graphics*, vol. 2021, pp. 1–18, 2021.
- [14] E. Akhtarkavan, B. Majidi, M. F. M. Salleh and J. C. Patra, "Fragile high capacity data hiding in digital images using integer-to-integer DWT and lattice vector quantization," *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 13427–13447, 2020.
- [15] Z. Su, W. Li, G. Zhang, D. Hu and X. Zhou, "A steganographic method based on gain quantization for iLBC speech streams," *Multimedia Systems*, vol. 26, no. 2, pp. 223–233, 2020.
- [16] Z. Huang, X. Zhang, L. Chen, Y. Zhu, F. An *et al.*, "A vector-quantization compression circuit with on-chip learning ability for high-speed image sensor," *IEEE Access*, vol. 5, pp. 22132–22143, 2017.
- [17] A. Filali, V. Ricordel, N. Normand and W. Hamidouche, "Rate-distortion optimized tree-structured point-lattice vector quantization for compression of 3D point clouds geometry," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, Taipei, Taiwan, pp. 1099–1103, 2019.
- [18] M. H. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1078–1091, 2012.
- [19] W. Zhao, L. Wang and S. Mirjalili, "Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 388, pp. 114194, 2022.
- [20] S. M. Syfullah, Z. B. Zakaria, M. P. Uddin, M. F. Rabbi, M. I. Afjal *et al.*, "Efficient vector code-book generation using K-means and Linde-Buzo-Gray (LBG) algorithm for Bengali voice recognition," in *Proc. of the Int. Conf. on Advancement in Electrical and Electronic Engineering (ICAEEE)*, Gazipur, Bangladesh, pp. 1–4, 2018.
- [21] C. Tang, L. Luo, Y. Xu, G. Chen, L. Tang *et al.*, "Sequence fusion algorithm of tumor gene sequencing and alignment based on machine learning," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–16, 2021.
- [22] P. Rani, P. N. Singh, S. Verma, N. Ali, P. K. Shukla *et al.*, "An implementation of modified blowfish technique with honey bee behavior optimization for load balancing in cloud system environment," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–22, 2022.