Hindawi
Security and Communication Networks
Volume 2022, Article ID 8125494, 11 pages
https://doi.org/10.1155/2022/8125494

Research Article

Empirical Compression Features of Mobile Computing and Data Applications Using Deep Neural Networks

Hana Almagrabi,[1] **Abdulrhman M. Alshareef**,[1] **Hariprasath Manoharan**,[2] **Hana Mujlid**,[3] **Ayman Yafoz**,[1] **and Shitharth Selvarajan**[4]

[1]*Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*
[2]*Department of Electronics and Communication Engineering, Panimalar Engineering College, Poonamallee, Chennai, India*
[3]*Department of Computer Engineering, Faculty of Computer Engineering, Taif University, Taif, Saudi Arabia*
[4]*Department of Computer Science, Kebri Dehar University, Kebri Dehar, Ethiopia*

Correspondence should be addressed to Shitharth Selvarajan; shitharths@kdu.edu.et

Received 20 September 2022; Revised 28 September 2022; Accepted 6 October 2022; Published 23 November 2022

Academic Editor: Praveen Kumar Donta

Due to the enormous data sizes involved in mobile computing and multimedia data transfer, it is possible that more data traffic may be generated, necessitating the use of data compression. As a result, this paper investigates how mobile computing data are compressed under all transmission scenarios. The suggested approach integrates deep neural networks (DNN) at high weighting functionalities for compression modes. The proposed method employs appropriate data loading and precise compression ratios for successful data compression. The accuracy of multimedia data that must be conveyed to various users is higher even though compression ratios are higher. The same data are transferred at significantly higher compression ratios, which save time while also minimizing data mistakes that may occur at the receiver. The DNN process also includes a visible parameter for handling high data-weight situations. The visible parameter optimizes the data results, allowing simulation tools to readily observe the compressed data. A comparison case study was created for five different scenarios in order to confirm the results, and it shows that the suggested strategy is significantly more effective than existing methods in roughly 63 percent of the cases.

1. Existing Approaches: A Survey

The fundamental approach to the proposed analytical framework is designed after examining different existing methods that provide support for compression techniques in the mobile computing process. During the analysis process, the state of various representations that are related to data computing is checked, and the drawbacks of various methods are observed. The major reason for examining relevant works in the compression of data is that all drawbacks must be solved using an optimal detection process in the presence of an analytical framework. In [1], the data compression technique is processed using a coding technique where more amount of misperception occurs at input data weights. As the weights are increased, it is essential to check the error function in the compression stage, but at output units, separate error state functions are not represented. Even though adaptive technology is considered, the data compression procedures remain the same as like in encryption and decryption cases thus requiring an advanced methodology. Hence, a survey is made [2] by comparing all the possible techniques for compression, and the values are transmitted using a well-defined wireless network. But to compress the large scale, data wireless networks are not needed as cost-effective function is not assured thus coercing all the mobile computing users to choose identical output units. In order to prevent the identical output units, the data are scheduled with different congregation systems [3]; thus, the compression of individual data is processed in an effective manner. However,

individual data compression will increase the latency, and as a result, the entire time period of computation and storage increases. Conversely, to reduce the cost of data transmission, a common collection unit using a base station is represented with varying size factors.

Apart from data size, the quality of data is also tested using different schemes and types [4] where the data are stored only in the available storage space. In addition, the input bandwidth for compression is much lesser; thus, only valid computing blocks are considered. Due to such deliberations, the compression ratio must be higher for different data computing segments, but in real time, it is not possible to provide high compression ratios. As an alternative, record management systems are introduced [5] with the best computational approach, and better storage space is considered in this type of value selection technique. With the procedure of large storage space, more number of compressed data can be stored and evaluated, thus achieving effective data for encryption. Moreover, an effective tool is chosen to realize all security functions in a mobile computing system with defined standards thus converting all data to be stored in a compact format. Although different data storage techniques are present, it is essential to check the reliability of the data storage process using edge computing procedures [6]. Thus, an initial check is made using edge computing nodes by installing different compression characteristics where compression ratios are made only at tolerable limits. Nevertheless, a standard has been specified for mobile computing systems as the rate of compression; thus, it is not possible to define any threshold value in any time period. Even the method with threshold values undergoes great drawbacks on threshold values, in this case, hybrid suggestions can be made. The aforementioned edge computing process is again carried out as a modernizing tool with robust deep neural networks (DNN) [7] where a mathematical framework is established with several assistance from defined computing devices. As there is a need for a device model, the cost of implementation is increased with the improper deployment of neural networks.

Additionally, the energy Internet model is introduced for mobile computing data compression that incorporates a local area network to compress the basic data that are present in textual representation [8]. The process of using such local networks will cause damage to edge servers; therefore, a random probability technique is made for reducing the percentage of latency in the entire system. It is observed that the usage of random probability increases the margin of compression data with extra overhead points. After changing the selected process to automatic compression, Ada deep neural networks are added thus solving all complex representations of computing nodes [9]. This type of problem-solving method is termed as intelligent systems where the energy of the compression block is greatly improved with clear illustrations. As the clear format of multimedia data is represented, it is necessary to use multiple channels, but at the same time, storage of compressed data in channels must be reduced. Consequently, all major challenges are analyzed using differential equations in order to check the memory processing capability of both compressed and uncompressed blocks [10]. While providing solutions to the big storage unit, a series representation model increases the mass storage of data thus even data clusters are stored during the compression process. Furthermore, many researchers [11] established some models for defining hierarchies to be followed in the case of different mobile devices such as high compression coefficients with limited resource allocation. Researchers have used an improved interframe prediction algorithm for video coding to acquire better-reconstructed images [12]. But the hierarchical model does not provide a structural view with any analytical framework; thus, a separate algorithm is integrated. Even in recent times, the researchers have directed the research model using a deep neural network where the design is completely based on complementary mobile terminals [13–17]. If each terminal in the mobile nodes is stationary, then insignificant data filters will be removed from the system. However, the modernized compression characteristics do not support different data filter types; thus, the recombination of different patterns in filter-based structures is needed. Hence, filter-based structures are changed to quantization aware technique thus evaluating the possible technique of implementing it using recombination states [18]. The state of recombination is provided using residual network data set thus reducing the data size for more number of times. Due to such reduction cases, the accuracy of the entire model is increased with an increase in data errors, and clustered weights do not provide extreme support during this progression period. Thus, to solve all the observed drawbacks, an analytical model is framed, and it is described in Section 2.

1.1. Proposed Methodology. All the existing literature with recent works is based on providing different compression features to applied data in the entire system. But, at the same time, all types of data are not compressed with best output values where even if compression characteristics are applied, data are not transmitted to end users in an appropriate format. It is also observed that compression starts using high weight vectors without any visible parametric variable. Since visible parameters are not added in existing methods, the weight factor increases in exponential terms; thus, compression of data is not provided correctly which directly leads to more storage space and time required for transmitting data is much higher. Furthermore, the distance of data that needs to be computed at different stages is not provided in the case of recently developed techniques, but better compression characteristics are achieved.

Due to the above-mentioned research gap in mobile data, the compression proposed methodology is incorporated with zero compression features. Additionally, visible parameters are added; thus, it is much easier to identify the number of weight factors and compression stages with low computational nodes. Also, the major gap of error minimization is reduced as both input and compressed sequences are compared, and if any mobile computing or data error occurs, it is corrected within an allocated time period. Once the errors are corrected and if the accuracy of corrected data

is higher, then data are retransmitted to the receiver. During the aforementioned retransmission, the latency of compressed bits is measured, and it is necessary that the gain of transmission must be maximized at minimized latency. Moreover, the projected method is applied with deep neural network (DNN) where effective functions are achieved in all three defined units using low data weight representations.

1.2. Objectives. The technique of mobile data compression which is carried out using zero compression units focuses on both minimization and maximization problems and is applied to multiobjectives cases as follows:

(i) To maximize the entire data load, compression of all bits is processed at the same time period with the reduction in input weight vectors

(ii) Appropriate data ratio must be maintained at each node with maximization of data reproduction rate

(iii) The other objective is to reduce the time period of data transfer and errors in data transmission.

2. System Metrics

The major problem that is present in the data transmission technique using a mobile computing system is represented using high data where an analytical model of data is needed for appropriate representation. Moreover, it is necessary to compress the mobile computing data; thus, an analytical framework is framed with high compression characteristics. Most of the system metrics in the existing method are not framed with a proper functioning unit; therefore, the data size that is present at the initial stage is exactly the same even during the channel transmission stage. In order to avoid the above-mentioned circumstance, the mobile computing model is designed with a new storage technique for loading all the compact data bits which can be used at a later stage. This loading technique is represented using equation (1) as follows:

$$l_i = \max \sum_{i=1}^{n} c_b(i) + \tau_i, \tag{1}$$

where c_b indicates the total number of compressed bits, and τ_i represents the optimal bits to be controlled.

Equation (1) is formulated as a maximization function with respect to the objective function as high standard compression is needed. Therefore, the ratio of compression is measured using (2) as follows:

$$r_c(i) = \sum_{i=1}^{n} \frac{u_c(i)}{C_c(i)}, \tag{2}$$

where u_c, C_c indicates the number of uncompressed and compressed bits, respectively.

In order to achieve a better compression effect, it is essential to reproduce all the values which can be represented as reconstructed mobile computing data without any loss of functionality. Thus, the reproduction rate can be expressed using (3) as follows:

$$\rho_i = \max \sum_{i=1}^{n} \frac{CR_i - CR_n}{E_i - E_n}, \tag{3}$$

where CR_i, CR_n denotes the compression ratio of ith and n^{th} bits, respectively. E_i, E_n represents the percentage of error for both i^{th} and n^{th} bits, respectively.

Equation (3) indicates the maximization function where the difference in terms of percentage will be provided. But the compression ratio usually varies with an access point in the case of wireless mobile computing and ad hoc mobile computing; hence, the uplink data rate of computing needs to be formulated using (4) as follows:

$$\text{up}_i = \max \sum_{i=1}^{n} \delta_{\text{in}} \left(1 + \frac{tp_i * g_i}{t_{\text{CPU}}(i)}\right), \tag{4}$$

where tp_i describes the transmission power of mobile computing data, δ_{in} represents the input bandwidth, g_i denotes the gain of the computing channel, and t_{CPU} indicates the time for computation.

Equation (4) represents the third maximization objective function where the users will transmit the data using a local network; therefore, in this case, if both transmission power and bandwidth are reduced, then the compression standard is assured. But the potential characteristics of mobile computing nodes will be different as the latency period is present which is formulated using (5) as follows:

$$\text{latency}_i = \min \sum_{i=1}^{n} \alpha_{\text{trans}}(i) + \alpha_{\text{exe}}(i), \tag{5}$$

where α_{trans}, α_{exe} indicates transmission and execution time periods.

Equation (5) represents minimization function where the latency of compression is not represented. Therefore, compression characteristics of latency are measured using energy values which are represented using (6) as follows:

$$E_i = \sum_{i=1}^{n} \vartheta_i * \text{latency}_i, \tag{6}$$

where ϑ_i represents the power of mobile computing data compression.

If there is more demand in case of compression from different users at the same time period, then mobile computing nodes will undergo an accuracy measurement as represented in (7) as follows:

$$\text{accuracy}_i = \max \sum_{i=1}^{n} D_c - \gamma_c, \tag{7}$$

where D_c, γ_c denotes the decision of classified and true classification representations.

During classification representation, there is a possibility that error functions will be represented as some of the data is highly compressed; thus, more changes are observed. Therefore, the error functions can be represented using (8) as follows:

$$\text{Error}_i = \min \sum_{i=1}^{n} (I_i - \omega_i)^2, \tag{8}$$

where I_i represents computing input sequence, and ω_i denotes compressed sequence.

The complete system model that is described using various parameters and variables is used for multiple objective case studies with minimization or maximization problem where the objective function can be represented using (9) as follows:

$$\text{obj}_i = \min \sum_{i=1}^{n} \text{Error}_i, \text{latency}_i, \max \sum_{i=1}^{n} \text{accuracy}_i, \rho_i, l_i. \quad (9)$$

Both the minimization and maximization problems are not implemented at the same time period; thus, compression characteristics of data are free from errors and latencies at high accuracy values. The entire analytical framework is transformed into a reasonable representation for informal implementation as a loop formation technique. Thus, the converted equation model is integrated with the optimization algorithm which is described in subsequent sections.

3. Optimization Algorithm

Since the data are directly reproduced as a systematic behavior where the human interface is not present in the system, it is necessary to convert system representations that are better unstated by individuals. Therefore, in mobile computing data compression, it is necessary to understand different patterns of input data before and after compression. The varying changes are directly marked in the compressed output; thus, all common problems are deciphered. Therefore, for understanding the computing characteristics in real time, a deep neural network (DNN) is integrated where the major advantage of predicting all complex data patterns is recognized using a three-layer process with a weighting function [19]. If the data in the mobile computing technique are compressed, then more amount of storage space is needed for storing both compressed and uncompressed data, but DNN uses only less memory space in order to store such data computing compressed functions [20–22]. As a result of low storage space, much quicker response is achieved in DNN as multiple layers in the system match both compressed and uncompressed data at same time periods. Furthermore, the resources that are allocated to DNN are much lesser; thus, even for data compression, the mobile computing system is designed at low cost factor. In addition, the number of parameters that need to be computed in DNN for compression measurements can be categorized using three factors that are expressed in mathematical terms using (9) as follows:

$$O_w(i) = \min \sum_{i=1}^{n} \frac{O_{\text{input}}(i) - K_i + P_0}{\mu_i}, \quad (10)$$

where O_{input} denotes input weight functions, K_i, P_0 indicates kernel and zero data compression functions, and μ_i represents number of data compressed pixels.

Equation (9) represents minimization of DNN functions with respect to input weight factors thus making the compression process to be much easier. In case if more weights are represented, then compression of data in the mobile computing process can be processed with a visible parameter which is represented using (10) as follows:

$$\text{weight}_i = \sum_{i=1}^{n} \frac{e^{-v_i}}{N_i}, \quad (11)$$

where e^{-v_i} denotes the exponential factor of the visible parameter.

By using the visible parameter, the compressed output can be expressed as a summation of weight parameters as given in equation (11), and the step-by-step implementation of DNN in mobile computing for compression is deliberated in Figure 1.

$$\text{output}_i = \min \sum_{i=1}^{n} \text{dist}_i + (w_1 + \ldots + w_n), \quad (12)$$

where dist_i indicates the distance of mobile computing data (Algorithm 1).

The process of DNN that is integrated with a defined system model is used for reducing the size of data without any modifications in represented data characteristics; thus, as a result, most of the digital industrial units can able to transmit a large amount of data at short time periods.In industrial process, the concept of big data is applied at both hardware storage capacity and at network bandwidth. The above-mentioned maximization process is provided when DNN is implemented as DNN provides storage space only to required units out of all available units in the system. Additionally, the computing resources are supplied at an appropriate amount even if intensive compression tasks are provided in the system.

4. Results and Discussions

This section provides real-time outcomes for mobile computing data by using some stored data set that is directly moved from cloud systems. At the initial stage, the experimental case studies are carried out using much low size data, and once initial checks are completed, big data are taken into account. Therefore, special compression standards are set for mobile computing data thus making the compression block to function more effectively for both big and small data sets. The data bits that are taken are converted to optimal form where both compressed and uncompressed bits are separated as there is a need to ascribe the compressed bits in case if needed. Thus, the above-mentioned process is carried out using a collection management system (CMS) that provides a better compact for file storage. Furthermore, the type of data that is present for experimental verification deploys both text and image sets using benchmark functions. Therefore, exact outcomes are achieved for defined formulations using a loop function where the programming model is incorporated using a mobile computing tool in MATLAB. Hence, real-time visualization graphs can be observed which makes the user to decide the amount of compression that is needed for further processing by using computing layer techniques. In addition, for examining the

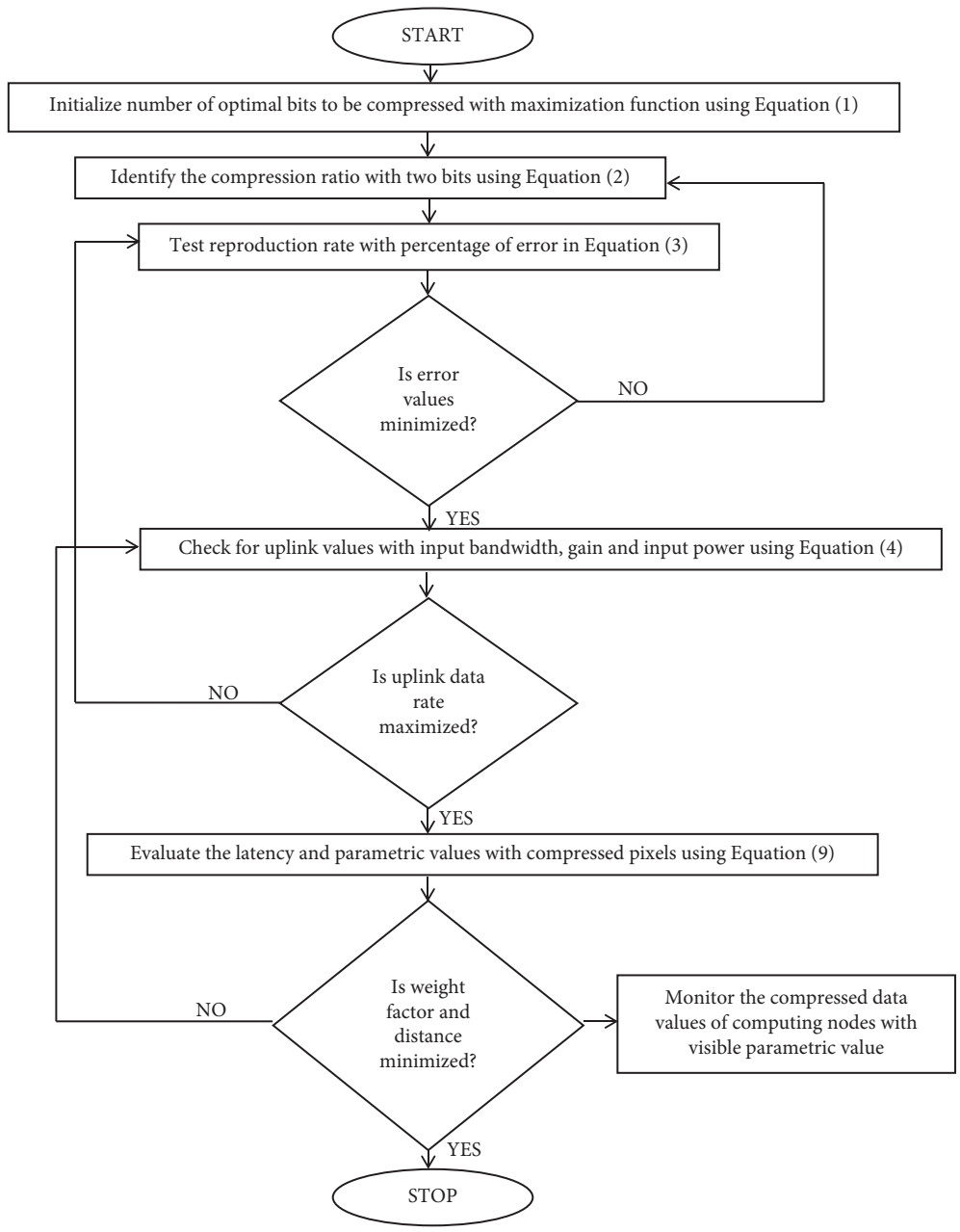

Figure 1: Flow chart of data compression for mobile computing.

outcomes, the proposed method computes the loss results using the defined scenarios as follows:

Scenario 1: analysis of data loading

Scenario 2: ratio of compression

Scenario 3: data reproduction rate

Scenario 4: uplink data rate

Scenario 5: determination of weighting functions

All the above-mentioned scenarios are carried out using lower-level functionalities only at the network layer. Whereas the higher level functionalities are considered in secondary mode only if high compression is needed. But in most cases, higher layer compressions are avoided as the data must be visible to users for more than 65 percentages; hence, more loss of data is evaded in all conditions. The detailed discussion about different scenarios is as follows.

4.1. Scenario 1. The process of loading data directly into the compression block is examined in this scenario where a number of data that are required for compression is also analyzed and stored. One major reason for providing simulation outcomes to this scenario is that more number of bits cannot be occupied in a particular system as the compression stage will take more amount of time for processing data that are present with low data size. Therefore, in the proposed method, only optimal bits are considered thus establishing a compact data compression model. Furthermore, the number of compressed bits is observed and added

Input: Initialize the total number of bits to be compressed with optimal bit representation and maximize the loading units $C_b\,(C_b \le i \le n)$, $\tau_i\,(\tau_i \le i \le n)$ and input weight functions of all mobile computing nodes;
Output: Optimized compressed data pixel values for processing mobile computing data and compressed ratio using deep neural network procedures at much shorter time periods and good expected rates;
Step 1: At first, the objective function is constructed with the uplink data rate using up_i;
Step 2: Initialize the transmission power to multiple data packets for compression with input bandwidth δ_{in} that must be followed by computational time periods $t_{\mathrm{CPU}}\,(i)$ with $0 \le i \le 1$, and its gain values g_i with compression periods of mobile computing process;
Step 3: While $(\mathrm{up}_i < \mathrm{threshold}_i)$ do.
Provide the compression ratio $r_c\,(i)$ in each bits separately for storing uncompressed and compressed data in a systematic way for computing the number of data compression loads in mobile computing automation process by using (2);
Verify the compressed ratio of ith, n^{th} bits and compute percentage of error E_i, E_n with comparison case using compression data separation values by using (3) for identifying the critical changes;
If the uplink data is higher up_i is not at $(\mathrm{up}_i < \mathrm{threshold}_i)$ do
Modify the time and gain for computation using maximization framework as represented using (4) that is having input bandwidth rates δ_i with $1 \le i \le N$ into N number of computing states;
//Weight setup
Update the input weight values $O_w\,(i)$ with kernel values and zero padding measurement function K_i, P_0 by generating the low compressed pixels μ_i using as shown in (9);
//Compressed data measurement
Select the output unit with different weights using distance measurement function dist_i as defined in (11);
Update the information about visible parameters in the computing network using (10) with separate weighting information values followed by the error function representation Error_i and compute the new latency period E_i as defined in (6);
The improvements in minimization of errors are represented using both compressed and input sequence with difference in compressed data which are updated by using (8);
$w_{\mathrm{new}} = w_{\mathrm{old}} + 1$;
End;
Step 4: If $(g_i < 0)$ then
$\mathrm{up}_i \leftarrow 0$; //Interchange the existing solution in the current loop with the new solution;
End if;
Step 5: If $(\mathrm{Error}_i\,[0, 1] < 1)$ then
Re-initialize the compressed mobile computing data with new segments;
Obtain the overall best solution;
End if;
Step 6: If $(g_i < N)$ //Existing solution is replaced with the new solution
$w_i = w_{\mathrm{modified}}$;
$\mathrm{up}_i = N$; //Attain the most feasible solutions for determining the overall best solution;
Increment the count w_i by 1;
Return the best overall solution;
End;

Algorithm 1: Deep neural network (DNN).

with optimal bits thus establishing total data bit representation. This type of bit representation is termed as the control stage as high data bits are converted in an automated manner; hence, functioning units are established appropriately without any data compression error. The total data compression bit values with the loading technique are simulated using Figure 2.

From Figure 2, it is pragmatic that total number of data bits is higher with 10000 to 30000 in varying step sizes of 5000 bits. In the projected method, step size is set as complex due to the incorporation of DNN in the system as some visible data factors are added. Due to much larger data bits, the compression block takes more amount of time for processing the entire data for computing. Hence, optimal data are considered from 3000 to 7000 at varying steps of 1000 bits where data complexity is reduced by removing unnecessary data in the system. After this removal process, compression of the bit period starts with a number of loads where the outcomes at the above-mentioned segments are compared with the existing method [6]. From the comparison outcome, it is observed that the number of loads for the proposed method is maximized with the original data whereas the existing method minimizes the original load data. Thus, at a later stage, compression is performed with less data which are much difficult to handle, and this can be verified with 25000 bits of original data where optimal bits are set at 6000. For the aforementioned data bit, the total load in the case of the proposed system is 3000 whereas, in existing method, it is equal to 2800 bits.

4.2. Scenario 2. In this scenario, the compression ratio of the total number of bits is made using the proper definition standard. The compression ratio in the proposed method is used for reducing more amount of energy in the system as higher data will extract much higher energy than expected.

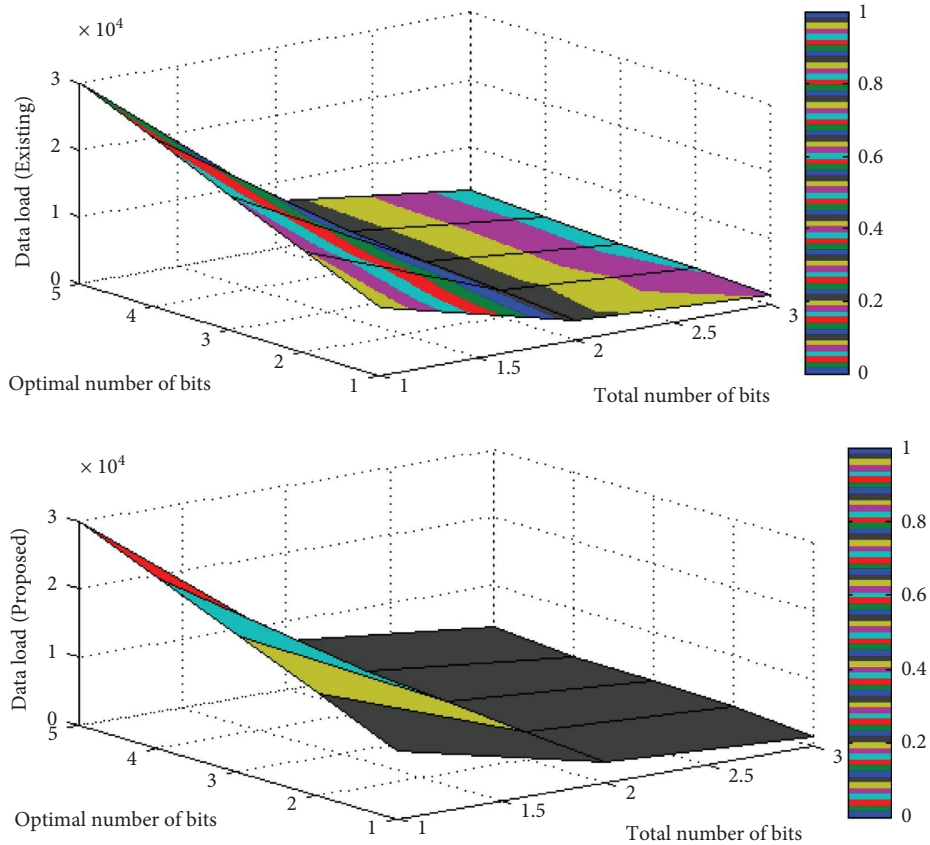

Figure 2: Comparison of data loads.

In many mobile computing systems, it is assumed that data energy is much higher only at the compression stage, but the exact cause is not accurate once the data are compressed. It is obvious that if a system provides more amount of uncompressed data, then the energy for transmission is much higher. Hence, a compression ratio procedure is made in the proposed method where all mobile computing users can transmit the data at low cost and storage. Moreover, the compression ratio is expressed using qualitative terms thus making DNN to function effectively in case of the large data set. In case if proper resources are not allocated for compression, then there is a need to allocate more resource set at the decompression stage. Figure 3 portrays the simulation plot for compression ratio with a comparison case.

From Figure 3, it is observed that a number of uncompressed bits are varied at improper step size as 300, 700, 1300, 1600, and 1900, respectively. Due to more number of uncompressed bits, the factor representation case is made with compressed bits as 100, 300, 600, 800, and 1000, and this stage compression ratio is examined and compared. From the comparison, it is much clear that the proposed method maximized the compression ratio without any data loss as compared to the existing method [6] with low compression values. This can be demonstrated using a number of uncompressed and compressed bits as 1300 and 600 where the compression ratio for the proposed method is 83 percentage which indicates that 1 : 4 data are compressed. But, with the same factor, the existing method provides 62 percentage as compression rate which is much lesser as 1 : 3 data are only compressed with loss of data. Hence, the compression ratio is much better in the case of DNN as compared to other algorithmic cases.

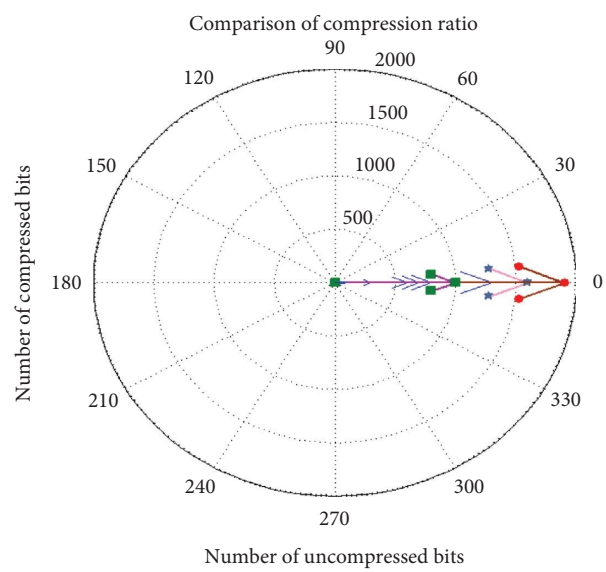

Figure 3: Expression of compression ratio.

4.3. Scenario 3. In real time, it is much difficult to design a mobile computing system that provides high compression with low error conditions. Hence, in the design of the transmission process, the reproduction rate is provided for all bit representation cases. The reproduction rate determines the difference between compressed ratios which are directly separated using percentage errors. Furthermore, this type of reconstructed data avoids high data loss in the system, and this is examined as a new case procedure in the proposed method. In a common mode representation, this type of reproduction rate is termed as normalization technique which is processed using feature extracted data set. But in the projected method, all information sources are parsed; thus, the entire mobile computing system establishes a common arrangement. As a result of common arrangements, multiple objective cases are processed at high computational speed thus maximizing the rate of reproduction at a low computational cost. Figure 4 provides the simulation outcome of the reproduction rate with a comparison study.

From Figure 4, it is realistic that the reproduction rate of the proposed method is maximized as compared to the existing method. For the verification case, the previous case compression ratio is considered in a common mode for both existing and projected methods as 73, 78, 83, 86, and 88 percentage, respectively. During this case, the percentage of error is much lesser as 2, 1.4, 1.2, 0.7, and 0.5, respectively, and with this low error percentage, the reproduction rate of the proposed system with DNN is higher. Even for a high compression ratio, the reproduction rate is maximized to 99 percent thus achieving full data end computing values. In the case of 78 percent compression ratio with 1.4 percent error, the proposed method provides reproduction rate as 98 percentage whereas the existing method provides only 84 percentage as a reproduction rate. Thus, this scenario proves that DNN is capable of reproducing all mobile computing data even at high error representation.

4.4. Scenario 4. The uplink data rate of mobile computing data is calculated in this scenario using transmission power and bandwidth parameters. Furthermore, the uplink data decide whether transmitting nodes provide highly effective operations as every mobile computing data node will be compressed at the transmission phase. Even when the data are transmitted to the receiver, the channel will compress the data; thus, to pass the data to the receiver, it is necessary to have necessary central processing unit time periods. Therefore, the transmission power and gain will be reproduced with separation values of time periods; hence, the compression stage provides a better data rate as compared to other systems. The total separated values are further reproduced using input bandwidth thus reducing the latency periods at both transmission and execution periods. Also, the power of computing nodes must be increased to make the visible parameter reproduced from the entire system; thus, as a result, uplink data are maximized. Figure 5 deliberates uplink data and its comparison with the existing method.

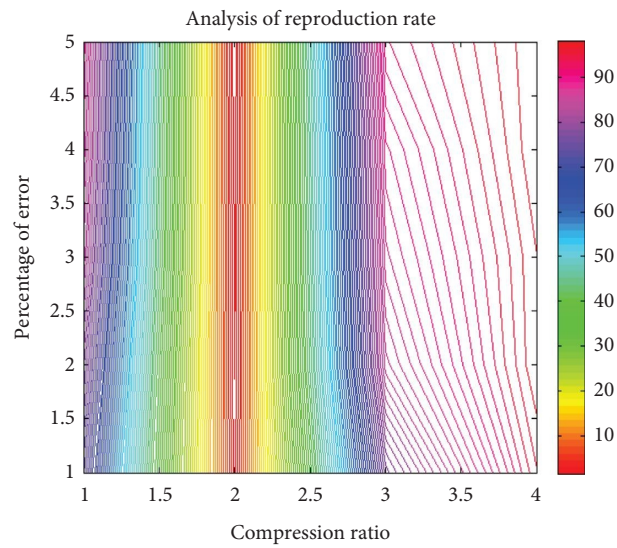

Figure 4: Rate of reproduction.

From Figure 5, it is observed that the transmission power that is provided for a compression block is 2.33, 3.45, 4.67, 5.21, and 6.03, respectively. For given transmission power, the gain of input data that are achieved in the case of full transmission signals is 52, 54, 58, 63, and 65 where time period of the uplink is 3, 5, 8, 14, and 16 seconds; thus, the reproduction process provides maximization of uplink data in the system. Once the initial specifications are completed, then uplink data speed is observed and compared with the existing method [6]. From the comparison, the uplink data that are provided by the proposed method are much higher for all transmitted periods even at a high compression ratio by the transmitter. But with the same data set, the existing method provides much lower uplink data rate, and this can be verified with 5.21 milli watts of transmission power with 63 percentage of gain where uplink data are provided at 17 MHz and 33 MHz for existing and projected methods, respectively.

4.5. Scenario 5. The process of weighting functions is determined in this scenario using kernel and padding parameters. If the computing data are transmitted, then it is essential to provide much low weighting functions to all defined compression blocks. But if the weight of the data is higher, then both accuracy and error will increase; thus, a visible parameter is represented using an exponential function. In addition, the data-compressed pixels are considered as additional weight functions even though it is discarded completely from the system. In case if individual weight functions are increased, then total representation with respect to distance will be maximized. Hence, the proposed method is introduced with DNN to overcome the above-mentioned maximization problem. At the initial stage of compression, only low-weight factors are introduced in the projected system thus making the conversion process to be much easier. Figure 6 provides a simulation analysis of weighting parameters with the comparison case study.

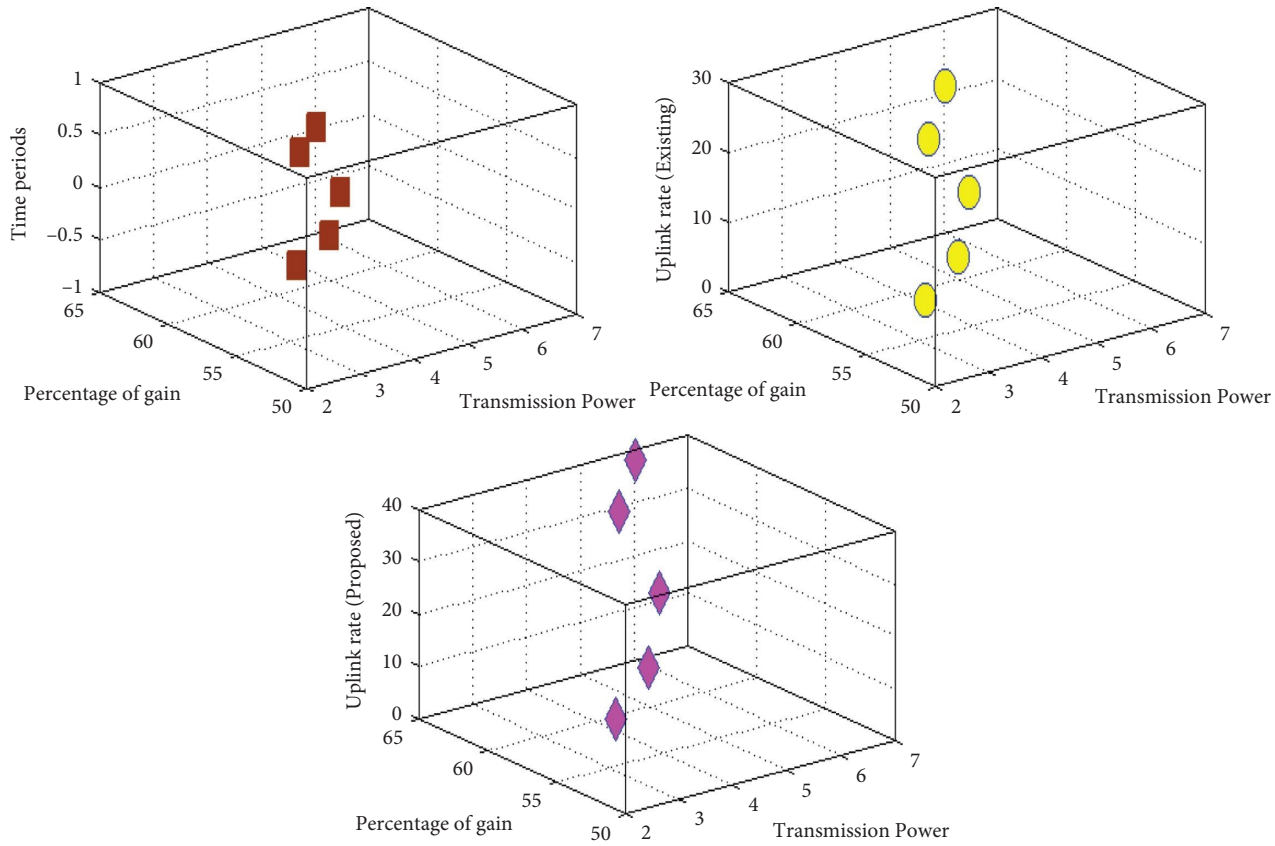

Figure 5: Time periods of transmission.

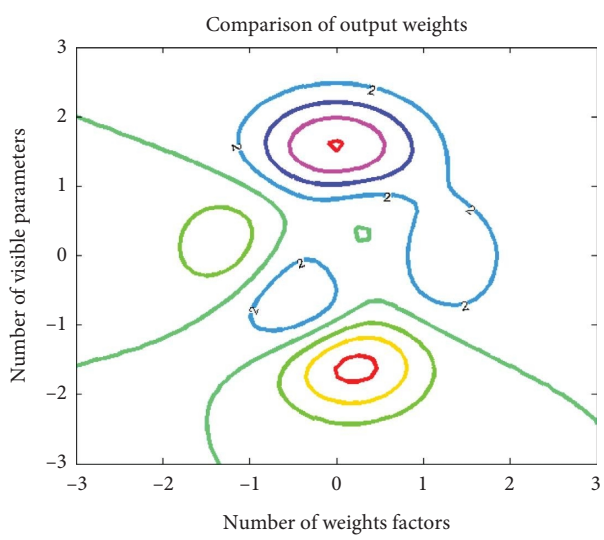

Figure 6: Computation of output weights.

From Figure 6, it is realistic that the number of weight factors is varied in step size of 8 up to 40 units. During this change, a number of visible parameters are varied in step size of 2 up to 10 units where due to this variation, the output units are varied with distance measurements. From the output measurement, a comparison case is made thus making the output unit to be much lesser than threshold values. Also, the existing method [6] increases the weight of the defined function as the distance of computing data is maximized. This can be verified with a weight factor of 24 and visible parameter as 6, and in this input, the weighting data set output unit is equal to 8.54 and 1.75 units for existing and proposed methods, respectively. The above-mentioned big difference in weighting function is due to the incorporation of DNN in the case of the projected method.

4.6. Robustness Characteristics. In the process of mobile compression, foremost importance is provided to the allocation of bits to each mobile node in the entire network. If the allocated bits are much lesser than the original representation values, then compression techniques will be robust as storage required for data during transmission and reception stages is minimized. Furthermore, the data which are converted into useful information must boost the performance of entire network bits, and if it is not achieved in a real-time scenario, then the system is indicated as highly robust, and it cannot adapt to any changes in the network. Additionally, entire system errors must be minimized with low disturbance in network signals; thus, if high deviations are present, then the entire system is highly robust to small changes in network structure. The robustness characteristics of mobile data compression are illustrated in Figure 7.

From Figure 7, it is observed that a comparison case study is provided to prove that the robustness of the proposed method is minimized than the existing approach. This can be demonstrated using ten different iteration values

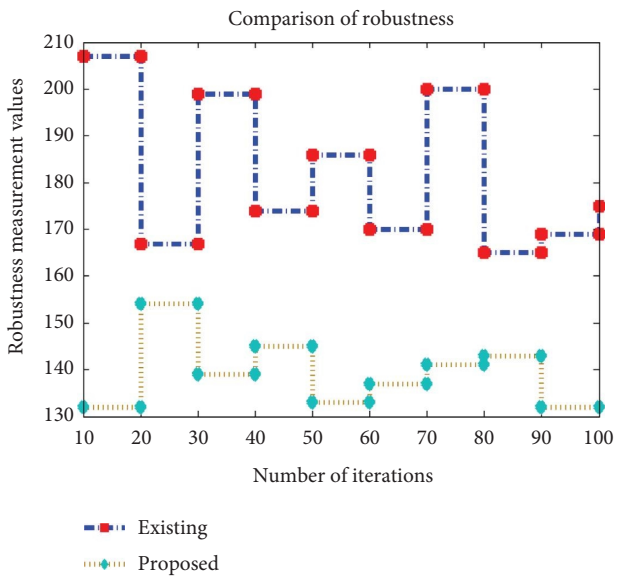

Figure 7: Robustness characteristics.

from 10 to 100 where for each iteration value, the robustness of both existing and proposed methods is provided with the same data set compression measurements. As a result of robust compression measurement, the proposed method minimizes and maintains the same robust values even at high iteration states. This can be proved using iteration periods of 10, 90, and 100 where robustness remains constant in the case of compression terms for the projected approach whereas the existing approach provides the change in robustness values of about 207, 169, and 175, respectively.

5. Conclusions

The problem of compressing high data that is present in mobile computing nodes is solved using DNN with a new analytical model where storage space for computing data is increased. In existing cases, many methods are represented by increasing the storage space of data, but the computing process involved in such cases using video and audio segments is much lesser. Thus, in the proposed method, analytical determinations are made for all common data segments in mobile computing systems with optimal threshold values. The bandwidth of other computing systems are comparatively lesser since the data transmitted in different forms quite often fails to reach the receiver at appropirate time period. But the above-mentioned case is not applicable for mobile computing nodes as data in the computing process are transmitted with low bandwidth; thus, only optimal bits are considered in the representation. Since optimal bits are represented, the compression ratio in the projected model is much higher with the same transparency values in all images. Therefore, as a result, the loss periods are reduced with the maximization of the accuracy factor by implementing visible parameters and weight factor in the system demonstration model. Furthermore, the errors during the data transmission phase are highly reduced as a collaborative server-edge mobile computing model is designed. To examine the integration of analytical framework with optimization algorithm which is carried out using DNN, five scenarios that include a loading of data, compression, reproduction, uplink data rates, and input weight determinations are tested and simulated using MATLAB. Furthermore, the simulation setup is made in real time and compared with the existing approach where the outcomes of the proposed method are much more effective even after data compression for about 63 percentage. In the future, the proposed method can be implemented with better compression standards by modifying the input with additional bits and even high-weight factors can be considered.

5.1. Policy Implications. The major consequences of processing mobile compression using DNN are that in the case of medical diagnostics, it is always required that even if the compressed images are represented in the same quality factor, there is a high-risk factor that small stem cells may be unexploited. Due to unexploited characteristics, the compression features might provide in accurate results with nearly 10 percentage changes in total data that are transmitted to the destination. Even in other applications, the consequence of mobile computing nodes will be highly robust; thus, extreme care must be taken to avoid such type of extreme compression.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Usama, Q. M. Malluhi, N. Zakaria, I. Razzak, and W. Iqbal, "An efficient secure data compression technique based on chaos and adaptive huffman coding," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 2651–2664, 2021.

[2] P. R. Kshirsagar, H. Manoharan, H. A. Alterazi, N. Alhebaishi, O. B. J. Rabie, and S. Shitharth, "Construal attacks on wireless data storage applications and unraveling using machine learning algorithm," *Journal of Sensors*, vol. 2022, pp. 1–13, Article ID 8457116, 2022.

[3] J. Berlińska, "Scheduling for data gathering networks with data compression," *European Journal of Operational Research*, vol. 246, no. 3, pp. 744–749, 2015.

[4] 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.

[5] J. Li, "Design of an Effective Archive Management System with a Compression Approach for Network Information Technology," *Wireless Communications and Mobile Computing 2022*, vol. 2022, Article ID 032017, 2022.

[6] S. Lu, Q. Xia, X. Tang, X. Zhang, Y. Lu, and J. She, "A reliable data compression scheme in sensor-cloud systems based on edge computing," *IEEE Access*, vol. 9, pp. 49007–49015, 2021.

[7] Y. Yan and Q. Pei, "A robust deep-neural-network-based compressed model for mobile device assisted by edge server," *IEEE Access*, vol. 7, pp. 179104–179117, 2019.

[8] M. Thirumalaisamy, S. Basheer, and S. Selvarajan, "Interaction of secure cloud network and crowd computing for smart city data obfuscation," *Sensors*, vol. 22, no. 19, Article ID 7169, 2022.

[9] B. Thirumaleshwari Dev and S. Shitharth, "Multiple face detection using haar - AdaBoosting, LBP- AdaBoosting and neural networks," *IOP Conference Series: Materials Science and Engineering*, vol. 1042, no. 1, Article ID 012017, 2021.

[10] S. Liu, J. Du, K. Nan et al., "AdaDeep: a usage-driven, automated deep model compression framework for enabling ubiquitous intelligent mobiles," *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3282–3297, 2021.

[11] S. Götschel and M. Weiser, "Compression challenges in large scale partial differential equation solvers," *Algorithms*, vol. 12, no. 9, p. 197, 2019.

[12] S. Zhu, S. Zhang, and C. Ran, "An improved inter-frame prediction algorithm for video coding based on fractal and H.264," *IEEE Access*, vol. 5, pp. 18715–18724, 2017.

[13] R. V. Bidwe, S. Mishra, S. Patil et al., "Deep learning approaches for video compression: a bibliometric analysis," *Big Data and Cognitive Computing*, vol. 6, no. 2, p. 44, 2022.

[14] J. Han, "Texture Image Compression Algorithm Based on Self-Organizing Neural Network," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[15] Bo Wang, Y. Liu, G. Shou, and Y. Hu, "Energy consumption minimization using data compression in mobile edge computing," in *Proceedings of the IEEE/CIC International Conference on Communications in China ICCC*, Chongqing, China, Auguest 2020.

[16] T. Liang, S. Shi, Y. Tao, and B. Zhang, "Research on deep learning model design technology of mobile terminal," *Journal of Physics: Conference Series*, vol. 2303, no. 1, p. 012031, 2022.

[17] M. Zhao, X. Tong, W. Wu, Z. Wang, B. Zhou, and X. Huang, "A novel deep-learning model compression based on filter-stripe group pruning and its IoT application," *Sensors*, vol. 22, no. 15, p. 5623, 2022.

[18] B. Predić, U. Vukić, M. Saračević, D. Karabašević, and D. Stanujkić, "The possibility of combining and implementing deep neural network compression methods," *Axioms*, vol. 11, no. 5, p. 229, 2022.

[19] S. Shitharth, P. Meshram, and P. R. Kshirsagar, "Hariprasath manoharan, vineet tirth and venkatesa prabhu sundramurthy,' impact of big data analysis on nanosensors for applied sciences using neural networks," *Journal of Nanomaterials*, Hindawi, vol. 2021, Article ID 4927607, 2021.

[20] A. Scaglione and S. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," *Wireless Networks*, vol. 11, no. 1–2, pp. 149–160, 2005.

[21] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," *Frontiers in Artificial Intelligence*, vol. 3, pp. 4–23, 2020.

[22] E. Arce-Medina and J. I. Paz-Paredes, "Artificial neural network modeling techniques applied to the hydrodesulfurization process," *Mathematical and Computer Modelling*, vol. 49, no. 1–2, pp. 207–214, 2009.