The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

Optimized Data Management Using Energy Efficient IOT Data Compression Framework with Edge Machine Learning

Sunil Chahal
Director- Technology Delivery, Concepts IT Inc.
messc0610@gmail.com

Abstract:
IoT devices generate huge data in a very short period of time and they need to be processed and the results to be produced within a time bound. The major challenge in this procedure is to send a large volume of data to the cloud and the power consumption involved in this. Sending large data for a long period of time will drain the battery of the IoT device quickly. One solution to this problem is to use edge devices instead of cloud for processing the data. The advantage of using edge devices over the cloud environment is the data will be processed near to the data where it is generated; it will reduce the time between submissions, processing the data and returning the results. In this paper an energy effective compression method is proposed so as to analyze the data using edge devices along with machine learning techniques. The dataset used here is Udacity self-driving car dataset which consists of 15000 97942 images of 11 classes. The image is applied with lossy compression.Here DCT lossy compression is used before it is transmitted. Next the compressed data should be reconstructed to be processed in the edge device. The supervised deep learning method (RCNN) with SZ is used for the reconstruction process. It is observed that this proposed method outperforms when compared to the existing techniques.
Keywords: IoT, Edge Device, DCT lossy compression, RCNN, Udacity self-driving car dataset

I. Introduction

Processing IoT data now requires the use of globally distributed, centralized cloud computing. The Internet of Things (CoT) with cloud assistance, however, encounters a number of challenges, including transmission delay, bandwidth limitations, and excessive energy usage. One example is the amount of energy required to transport a single piece of data over the cellular network, which reduces the lifespan of the IoT system. Edge computing, on the other hand, has become a promising paradigm that moves cloud services to the network's edge. It can be compared to a decentralized cloud that allows for local decision-making and moves computational power closer to the data source.

In many IoT applications, edge computing has shown to be a superior solution to the cloud [1]. For instance, because of the high latency and inefficient bandwidth generated by the numerous sensors connected to the network, applications that require near real-time answers, such as autonomous driving automobiles and eHealth, cannot function well with the cloud. The fundamental building blocks of IoT designs include Wireless Body Sensor Networks

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

(WBSNs),Wireless Sensor Networks (WSNs), , and wearable technology. Many of these intelligent things, which are in charge of data collecting, processing, and transmission, nevertheless rely on batteries and have limited resources. The microcontroller (MCU), transceiver, and sensor units are the three main components of a smart item that use energy. Data transfer is commonly acknowledged to be the IoT node task that uses the most energy overall [2] [3]. Transferring computational operations from the cloud to the edge is a significant step toward improving energy efficiency in IoT applications. In general, radio communication between IoT nodes and the edge uses less energy than sending data via cellular networks straight to the cloud [1] [4]. The lifespan of IoT nodes can be extended and storage space can be saved at the edge by minimizing the quantity of data that needs to be transferred there.

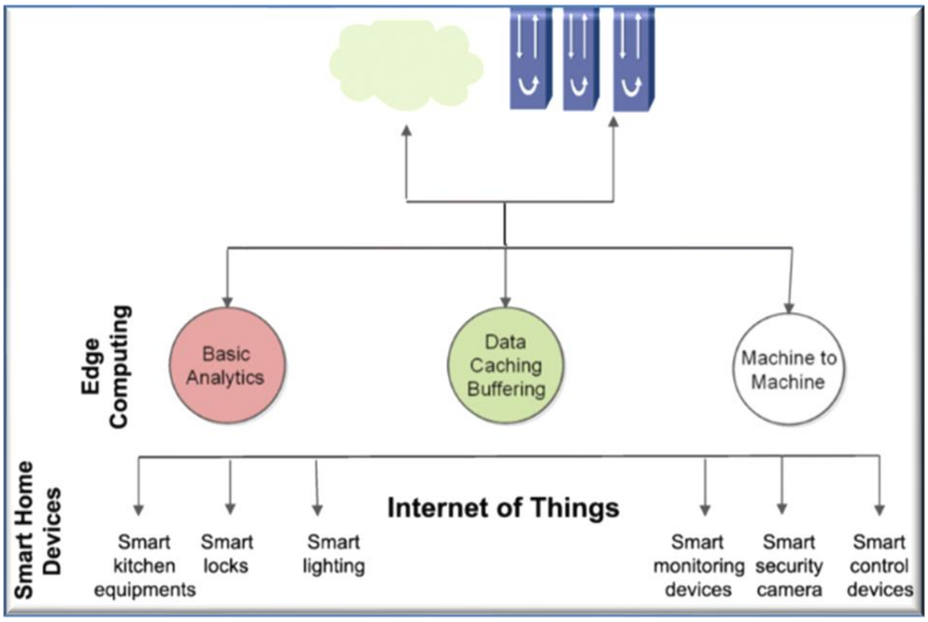

Fig 1: Internet of Things Network Architecture with Edge

The study look at the architecture of Internet of Things network architecture depicted in Fig. 1, where the data is periodically delivered to all edge node itutilisingthe minimus range message transfer protocols of items ,wearable smart devices with sensors (e.g. WiFi, Bluetooth). Processing, analysing, filtering, storing, and transmitting the data to the cloud are all the responsibility of the edge node. First, a quick error-bounded lossy data compression approach is proposed for use on IoT devices as a solution to the energy conservation challenge. The goal is to lower the amount of bits that must be sent to the edge node on a periodic basis.

The effectiveness of edge deployed ML and DL models trained on lower quality regenerated data than the source data. In order to do this, we take into account the use case of driving behaviour monitoring, in which edge nodes receive physiological data from drivers in order to gauge their degree of stress.

The issue raised in this work can be phrased in the following way: Does representation and data analysis on the cutting edge suffer from it's information loss caused by lossy compression and energy-saving algorithms used on IoT nodes? The remainder of the essay is structured as follows: The work on data reduction and edge analysis for Internet of Things

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514
applications is shown in part 2. The suggested compression technique is described in part 3. This situation examined dataset in this work is discussed in Section 4 along with a forecasting system used intthe edge in part 5. The investigation's findings are described in Section 6. The paper is concluded in Section 7.

II. Related Work

2.1. DL and ML on the Edge device

By utilising machine learning and deep learning techniques for data analysing, processing devices may become intelligent and use bandwidth and confidential more efficiently. The authors of [5] integrated DL for Internet of Things into the environment of edge computing environment and put forth a method that enhances user privacy while maximising network performance.

The conventional approaches that utilize on cloud computing can benefit from machine and deep learning in edge computing by:

- Processing data with typical machine learning methods and transferring the outcomes or essential characteristics retrieved from raw sensor data.
- Deploying a portion of deep learning network layers on the edge and transferring the features that were retrieved but lower in size than the raw data.
- Using accuracy-preserving, minimally sized neural networks at the edge.
- Cloud-based network training and sending the edge-ready models.

By minimising the amount of data that needs to be sent to the cloud, the aforementioned strategies ease network load. A DL network, for instance, execution and reduces the volume of the produced attributes from the preceding stage. The number of the features to be communicated to the cloud will be smaller and more unintelligible as more layers are placed on the edge, boosting privacy.

Mobile phones, IoT gateways, and local PCs are examples of edge nodes, which have less computing power than cloud servers. On these devices, neural networks should be of a decent size. The authors of [6] demonstrated how various algorithms and low weight libraries and algorithms may installed an edge nodes like cellphones to establish real-world data processing.

2.2. Reduce the volume of data in Internet of Things Systems

Developing a system that captures the occurrence being felt & installing on Internet of Things enabled devices and edge node is how the suggested method operates. The benefit of prediction techniques is that, the model at the edge forecast the observed parameter without requirement for radio communication unless the validation loss transcends a specific threshold. Such estimation algorithms, however, are limited by the high data collection frequency and rapid data variation of devices like high frequency motion sensors. Many aggregation and compression techniques have been presented [10, 11, 12, 13, 14] that benefit from the temporal correlation in the collected data. A data screening method on the basis of Pearson coefficient metric was suggested from authors in [10]. This technique aggregates the metrics basis on the correlation among the categories after recursively splitting the dataset

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514
into two equal portions. A cumulative data strategy for thereceived data streams in the monitoring systems based on IoT was put forth in [11] by the author. The suggested approach uses an approximation with "extremums" strategy to lessen the amount of data that needs to be transferred or saved. The outcomes demonstrate that this technology was capable of compressing temperature data up to 10 times. The authors of [12, 13] suggested data compression methods that profit from the temporal correlation in the gathered data.

For the time series analysis based on the multiple variablesreduction in Internet of Things applications, the Compressive Sensing (CS) and transform domain reduction methods are often employed for the pictures also been developed. The authors of [15] suggested a multi-signal reduction method on the basis of fuzzy transform principle. The multi stage signal surrounding data gathered by a network which is based on wireless system was subjected to the proposed approach, which resulted in a data reduction of almost two times. The 2-D lifing wavelet transformation was suggested by the authors in [16] as a method of compressing multi-signal datum gathered on several IoT devices. With help of the Haar wavelet, the suggested approach gets the reduction proposition of 1.32 & a recovering precision of 98.3%.Methods that compress data in the transform domain can accurately recovering the data. The reduction capacity of such approaches is still somewhat constrained. Meanwhile, in recent years, CS theory has developed as a successful strategy for low power consuming in the applications powered by IoT [17] [18]. CS techniques ensure the precise signal retrieval fromsample signals at a considerably below the conventional Shannon-Nyquist theorem. This is done by taking use of the signal sparsity. However, CS methods struggle while handling with the non sparse signals with high dimensional signals that have a variety of characteristics and value scales. The two main shortcomings of the aforementioned theories are that they are not validated on actual hardware and produce low data reduction ratios on data with dynamic multistage sensor.

This study suggests a data reduction method for IoT hardware and software with limited resources which is effective with multivariate time series and may be used to actual wearable technology. The suggested lossy compressor is discussed in the parts that follow, along with the effects that comes with data analysis is the loss of data at processing system.

III. Proposed Compression Technique

3. Error Constrainedlossy compression

The condensed form of a material offered along with [19] is provided with this paper. For High Performance Computing (HPC) applications, SZ, a quick error-bounded lossy compression method, was proposed by the authors in [19]. To cope with the enormous volumes of data created and handling the execution of HPC systems operations, this compression approach has been devised. The original SZ reduces the size of loading raw data in the form of input stream, which may contain various data types and forms (both single and double precisions). We suggest here that the SZ method be modified for Internet of Things (IoT) devices by just taking into account decimal point data types as well as rejecting all other data types, results in reduced script sizes and makes it simpler to assemble on small systems. Additionally, the procedure here was modified such that it can now accept a one

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

dimensional array of decimal sensing data as input and output a byte array will be sent to the processing node.

In the following are the reasons for selecting SZ for IoT applications:

- Multivariate time series with various attributes and scales can be compressed using SZ.
- SZ permits the use of an error bound to control information loss.
- Compared to multi-dimensional transform domain compression approaches, SZ results in a greater compression ratio.

Algorithm 1 provides a definition of the suggested compression strategy. After each interval P of time t, it is assumed that data are sent to the edge. The gathered information is presented as a X Y array, where X stands for the no. of readings and Y for the no. of characteristics.Take a non static sensor that after a period P, gathered 128 data from its sensors like accelerometer and the gyroscope for its 3 axes. M then equals 128 and N then equals 6.

The 2-D array is first transformed into a one dimensional array (Refer Line 4 of the Algorithm 1).

This flat array isis further reduced by the SZ method which is lossy. The array of binary digits that results is then sent to the processing device. This essential measure is taken with the modified SZ reduction algorithm are shown in Algorithm 2. It should be noted that the adaption was completed by removing the required features of the original SZ in order to put it on the body sensors and the gadget with limited resources.

Algorithm 1 Proposed compression scheme Require:

E (error bound)

1: while Energy > 0 AND Sensors status = ON do

2: for each period do

3: data[M, N] ← collected sensors data

4: input[M × N] ← Flatten(data)

5: bin output ← adapted SZ(input, E, M, N) (Alg 2)

6: transmit Data(bin output)

7: end for

8: end while

Proposed edge

Requirement: input (One Dimensional array), E (error constraint),X (Number of rows),Y (number of columns) Ensure: output (array of binary digits)

1: Bestfit Curve-Fitting reduction

2: ReducingDynamic Data

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

IV. Case Study

Over the past few years, there has been a lot of interest in the intersection of IoT, cloud computing, and healthcare. The following are a few of the biggest obstacles facing healthcare applications:

- Communication lag caused by the cloud and Internet of Things devices
- Due to the volume of data collected, there is a limited network bandwidth.
- Security and privacy challenges are expensive.

Merits of edge computing manifest themselves in this area. Edge computing, as discussed in earlier sections, can be viewed as a significant result for response time and bandwidth problems in Internet of Things and healthcare technologies and its applications. Comparable to this, effective on-node data compression can lengthen the application's life and reduce the volume and number of transferred packets, which can have a favourable impact on latency and network capacity. However, for healthcare applications, data integrity and information loss via compression continue to be major problems. This section examines the effects of energy efficiency and data compression during the evaluation of the medical image data at the processing node. Let us take the example of monitoring driving behaviour and stress detection, where physiological signals are continuously gathered from the driver and sent to the edge node for analysis and stress level detection.

The dataset used in our research is described next, followed by a discussion of the data's processing and analysis, and lastly a presentation of the model used to determine the stress level.

4.1. Dataset

This research uses the Stress Recognition in Automobile Drivers database that was published on PhysioNet [20][21]. This dataset includes a variety of physiological signals captured from healthy volunteers while they drove along a predetermined path through and around Boston, Massachusetts. Each driver's driving duty is broken down into six portions and lasts anywhere from 50 minutes to 1.5 hours.

• Rest [1,6]: "Low stress" rest intervals are those that occur at the start and finish of the driving task.

• City [2,5]: Since the participants frequently dealt with traffic conditions, unforeseen situations caused by bicycles, and jaywalkers while driving down a busy main street, driving in urban areas is classified as "high stress."

Highway [3,4]: Periods spent driving on the highway are classified as "moderate stress." It should be noted that drivers' self-reported surveys in [20] have corroborated the data's categorization.

Since not all driver files contain the marker identifying the driving portions (rest, city, highway), this study only included 9 of the 17 drivers who were available in the database. Galvanic Skin Reaction (GSR), Electrocardiogram (ECG) and the Heart Rate (HR) from the body parts particularly the foot, the hand and the exhalation and inhalation rate are selected as the five physiological markers for stress detection.

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

4.2. Data pre - processing and examine

The noise in the HR and GSR signals has been eliminated for each driver data file as shown in [22]. In addition to the skin conductance level [SCL] and skin conductance response, the two major elements of the GSR waveform, duration and bandwidth characteristics are often extracted from the Electrocardiogram pulse, to explore effects on the data reduction on the message within the data (SCR). Comparing the attributesretrieved from the raw data with those extracted from compressed data is the goal of this stage.

By using time and frequency domain analysis techniques such FIR-filters and fast fourier transform methods, Table 1 shows the attributes that were retrieved from the signals received Electrocardiogram. As the continuation of convex optimized method suggested at [23], we extract the slow variation (SCL) and the quicker variations (SCR) from the GSR signal in order to examine it as a significant sensitive metric for emotional arousal. The feature based output is taken from the source of Galvanic Skin Reaction signal are displayed in Table 2.

Feature	Description
RMSSD	Inter-beat (RR) intervals' root mean square (the time intervals between consecutive heart beats)
meanNN	Mean RR distance
sdNN	RR interval for the standard deviation
cvNN	The ratio of standard deviation to mean deviation, or the coefficient of variation (CV),
CVSD	RMSSD divided by meanNN is the consecutive differences' coefficient of variation.
medianNN	Median of the successive absolute values of the RR interval differences
madNN	The RR intervals' median absolute deviation (MAD)
mcvNN	MCV, or the ratio of madNN to medianNN, is a measure of variation based on the median.
pNN20	The proportion of all RR intervals divided by the number of interval variations between subsequent RR intervals larger than 20 ms
pNN50	The proportion of all RR intervals divided by the number of interval variations between subsequent RR intervals larger than 50 ms

Table 1: Features extracted from ECG signal

Characteristics taken from the ECG signal summary of the Feature RMSSD Inter-pulse (RR) time gap root mean square the mean of the periods between successive cardiac cycle meanNN sdNN, the mean-RR distance RR interval mean difference cvNN The ratio between standard deviation to mean deviation, or the coefficient of variation (CV), CVSD RMSSD divided by meanNN is the consecutive differences' coefficient of variation. The consecutive relative readings of the Median-NN of the successive absolute values of the RR intervals' variations madNN. The inter-pulse time period Median Absolute Deviation (MAD) mcvNNMadNN divided by medianNN pNN20, or the Median Based coefficient of Variation (MCV), proportion of all RR intervals divided by the number of interval variations between subsequent RR intervals larger than 20 ms Divided by the total number of RR intervals are variations between subsequent RR intervals higher than 50 ms.

Feature	Description
meanGSR	Mean value of GSR signal
meanSCL	Mean value of SCL
slopeSCL	Difference between max and min values of SCL
meanSCR	Mean value of SCR
maxSCR	Max value of SCR

Table 2: Characteristics taken out of the GSR stream

4.3. RCNN - Regions with Convolutional Neural Network

The attributes listed in the Tables numbered 1 and 2 as well as Heart beat Rate and Respiration (Inhale-exhale) Rate are all taken into account in this section as an unified sequence to which a label is applied. A supervised sequence classification job is the prediction task. One input layer, one or more hidden layers, and one output layer make up an FFNN, which is a network with many neurons arranged in layers. The RCNN architecture utilised for the classification. There are 17 input neurons in the neural network, each of which corresponds to a sequence.

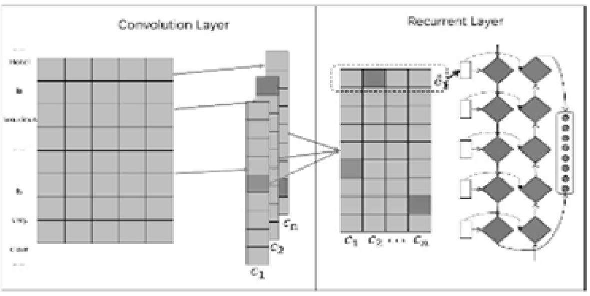

Fig 2: RCNN

17 Physical Characteristics:

Additionally, a correct label for that sequence is sent to the neural network. Specifically, if the sequence represents low, moderate, or high stress.This work utilizes aneural network

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

contains a minimal volume that retains precision, given that processing system needs a low weight method for execute reasonable ML [6]. The built network comprises four hidden layers with a total of 60 ReLU-activated neurons in them [24]. Quantity of stages of layers contained in the neural net was instructed using categorical cross entropy and the stochastic gradient descent "Adam" variant [25]. Furthermore, Ten-Fold Cross Validation been utilised to check as well as discover the ideal couple of parameters of the system, and the dropout regularization technique [26] been employed to avoid over training. This neuralnet is executed for 310 epochs and the learning rate is set to 0.01 to validate the model and identify the best set of hyperparameters. The network has now completed 300 epochs of training with a learning rate of 0.01.

V. Experimental Results And Analysis

The outcomes of using the aforementioned data compression approach on physiological data are shown in the sections that follow. The following two metrics—compression and the effect of message lost on forecasted precision—are discussed.

5.1. Metric compression and power efficiency

These drivers' signals and the data collection were captured at 496 Hz. Herethe data is loaded onto the external memory card of a wearable device in order to evaluate its effectiveness of the suggested reduction strategy. With the similar fashion, the Android NDK toolkit was used to build the compression approach written in C. The data array [X,Y] is sent using low power consuming Bluetooth technique (BLE) to the processing device, this is near to the personal computer, this is the place the data received are verified and evaluated, after each interval of p = 1 minute. The output is subsequently uploaded to the cloud. The no. of readings (X) is identical to 148900, and the no.ofcharacteristics (Y) is 5. 241 sessions, or around 4 hours, of the experiment have been completed. The disparity in the amount of data transferred with and without compression is depicted in Figure 4. Periods are indicated on the x-axis, and the no. of bytes required to refer the data in log measure is indicated on its y-axis. Around 2976000 bytes are needed for the transfer of the raw data after every period, whereas between 28732 and 41602 bytes are needed for the transfer of the compressed data. 103 times less data is transmitted as a result.

The wearable gadget is either collecting data continually (red line) or in an idle state (black line) (non static and heart beat sensors are in on position)

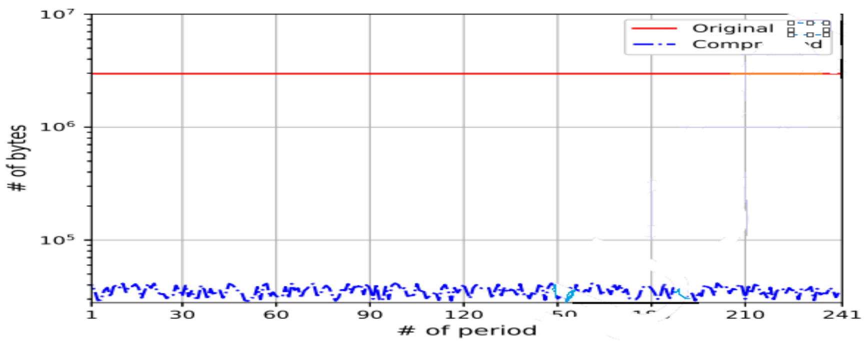

Fig 3: Volume of raw and reduced data transferred over the 241 periods

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

The line indicates by the yellow color is the device on the body which is used to extract the data and running the compression method after every period. The line indicated by the green color is thesame device is gather the data, doing compression, and transmitting it for every period. the line indicated by the blue line is that the same device is fetching the data, transmitting it, and running the compression algorithm after each period (no compression).The outcomes unequivocally demonstrate how data reduction affects communication task energy usage. After 241 periods, the chargeamount of the device that was gathering the data continually drained to about 86%. The device's energy usage can be observed to be slightly influenced by the sensing and processing tasks when compared to the idle state. On the other hand, transmitting the raw data drain the battery about 56%, while using the proposed data compression strategy before transmission reduced by the battery level to 83%. As a consequence, after four hours, the device life may rise up to 27%.

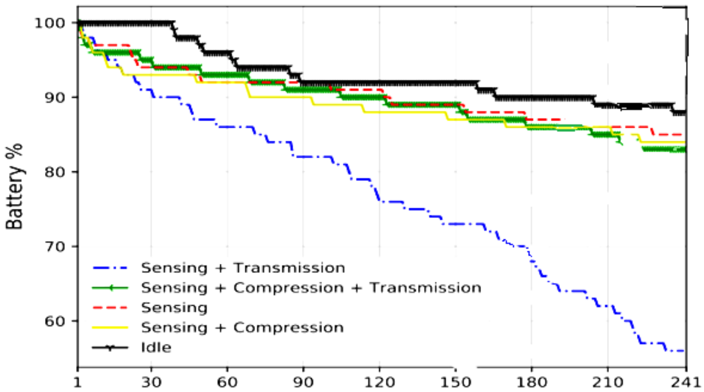

Fig 4: Polar M600 chargefor 241 periods

5.2. Missing of data and stress finding

Effect of data missing on the proposed RCNN's forecasting precision for stress level detection is examined in this section. A moving window of 30 seconds with a 75% overlayson the metric is employed to each of these driver datasets, and the physiological attribute mentioned in the preceding parts are retrieved for each window.

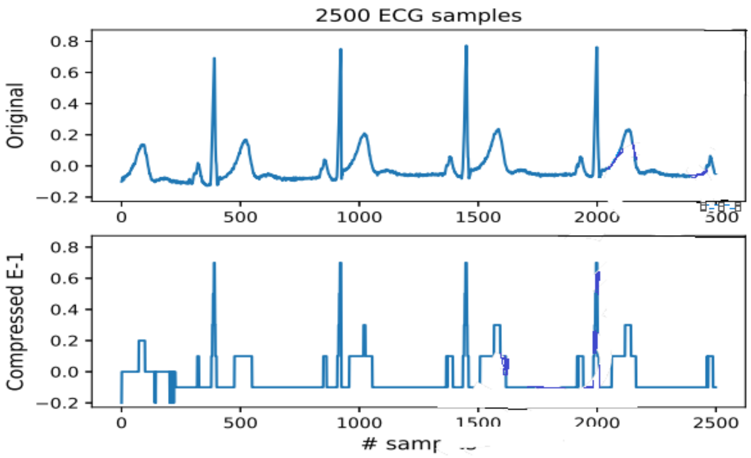

Fig 5: uses an error constraint of 101 to compare the raw ECG signal

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

Signal was transferred to the processing device with raw data but with the reduced data. The co-ordinates of the high positions are applied to get the data signal and also to retrieve the vital elements are not changed, despite the fact here is compression has somewhat modified the structure of the signal.For the nine drivers, Table 1 compares the Root Mean Square Error [RMSE] of the characteristics derived with the reduced and uncompressed GSR & ECG signals. Take note of how essential features like heart beat, inhale and exhale rate, and R-R gap have an average RMSE that is near to 0, indicating that the data reduction is very low of an effect on the information loss.

	Driver #									**Average**
	4	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**16**	
HR	0.2	0.08	0.07	0.10	0.04	0.08	0.07	0.07	0.26	0.11
RMSSD	1.57	1.68	1.18	0.94	1.20	2.45	3.77	2.28	2.42	1.94
meanNN	0.91	1.24	0.24	0.23	0.25	1.02	0.30	0.70	1.09	0.66
sdNN	0.61	1.01	0.29	0.33	0.58	1.12	1.65	0.87	1.44	0.87
cvNN	0	0	0	0	0	0	0	0	0	0
CVSD	0	0	0	0	0	0	0	0	0	0
medianNN	2.19	1.4	0.67	1.34	1.01	2.08	1.74	1.06	1.63	1.45
madNN	1.91	1.19	1.01	1.22	1.12	1.96	1.95	0.88	1.52	1.41
mcvNN	0	0	0	0	0	0	0	0	0	0
pNN50	1.78	202	1.23	1.51	1.91	1.92	2.24	1.56	1.27	1.72
pNN20	2.52	3.63	1.70	1.95	1.93	2.39	1.80	1.92	3.38	2.35
mean_gsr	0	0	0.01	0.02	0.02	0.01	0.02	0	0	0
mean_scl	0.23	0.22	0.23	0.21	0.21	0.20	0.20	0.19	0.23	0.21
slope_scl	0.38	0.36	0.34	0.34	0.28	0.33	0.34	0.29	0.44	0.34

Table 1: compares the properties retrieved from the uncompressed data and the properties taken from the rebuilt form, showing the RMSE.

The effectiveness of the aforementioned models at detecting stress levels is summarised in Table 2. The findings indicate that the two models' combined average accuracy is 98%. Due to the compression technique's denoising ability, it can be shown that the prediction accuracy was not only unaffected by compression but in some situations, such as for drivers 7 and 12, was even increased.

	Accuracy	
Driver	**Original**	**Compressed**
Driver 4	1.0	0.99
Driver 6	1.0	1.0
Driver 7	0.97	0.98
Driver 8	0.98	0.98
Driver 9	0.98	0.97
Driver 10	0.99	0.99
Driver 11	0.92	0.94
Driver 12	0.99	0.99
Driver 16	0.99	0.98
Average	0.98	0.98

Table2: shows the test sets' (25%) RCNN prediction accuracy for feature sequences taken from the two types of data (raw and compression).

VI. Discussion On Result

The majority of the devices linked with the internet available today have many sensors and can gather various types of data. Therefore, numerous device measurements at the single machine must be handled using the compression approaches. The relaxation time of this method should also short in real-time or near real-time applications for the need of feed the machine learning model to be trained with the data and provide the user with the grouping / organized results as soon as will. Even the techniques like compression sensing and the compression based on transformare approaches that may be widely used to deal along with thereadings of the multiplesensors, these techniques has a number of drawbacks that make RCNN the best option in these situations. For transform-based compression approaches to achieve an acceptable reduction rate, the raw data must first be converted into a set of coefficients. The coefficients must then be encoded using entropy. However, in order to accomplish a "accurate" reconstruction in CS, a signal must be sparse in some domain and noise-free, which is in contrast to the practical situations. Because the compression sensing is not a symmetric technique, it is ineffective for applications that call for quick responses because decompression requires more computational complexity than compression and takes longer to retrieve the data. Adaptability is another problem basis on transformation methods.

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 2499 - 2514

Alternatively, put reduction method have to be flexible additionally effective throughout the range of uses, topics, and pursuits. As an illustration, the time series analysis task which involves multiple variables utilized throughout this writing this dissimilar attributes for the distinct variables. Therefore, the individual variables of the data should match the requirements for compress sensing to function properly to fit compress sensing to the data with multiple variables, but it is not always true in this case.

Data size (bytes)	Compression time (seconds)	Decompression time (seconds)
11905000	0.026	0.03
3976000	0.022	0.05
973000	0.002	0.002
79200	0.008	0.001

Table 3: shows the proposed algorithm's typical compression/decompression times for various data sizes.

Table 3 displays how long it takes the proposed algorithm to compress and decompress input multivariate data of various sizes when it is running on a wearable device (Polar M600). As can be observed, SZ is suited for the mere real time systems because of the average compression/de-compression time is brief. The aforementioned factors, the suggested SZ may be a preferable alternative for the reduction of data fusion values due to that was quick compaction and convert it back to uncompressed format and the large reduction rate as well as the adaptability to various settings and implementation.

VII. Conclusion

Minimizing the volume of data which should be transmitted is a strategy to extend the lifetime of the edge IoT devices. Here the data is processed at the place very closer to the place where the data is generated that is the place of the device connected with internet. This technique reduced the latency time and also the power consumption for the transfer of the data. An energy-efficient technique for the reduction of the data strategy for IoT-Edge applications was put forth in this study. The proposed approach is based on an error bounded lossy compressor designed specifically for performing high computing environment which produce large amount of information while in operation utilizing multidimensional time-series data from the area of medicines the compression procedures was adjusted to operate with the polar M600 device. The result showed that by reducing the amount of data transmitted to the edge computing it may increase wearable lifespan by approximately 103 folds Additionally, we looked at the drivers stress recognition use case and investigated how the compression that involves loss affects the inspection, utilisation, and grouping of data related to medical field. The findings demonstrated the validity of the data recovered from compressed formats, and the classification accuracy attained after building the model using features from compressed data did not degrade.

References:

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (5) (2016) 637–646. doi: 10.1109/JIOT.2016.2579198.

[2] G. Anastasi, M. Conti, M. D. Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, Ad Hoc Networks 7 (2009) 537 – 568.

[3] M. A. Razzaque, C. Bleakley, S. Dobson, Compression in wireless sensor networks: A survey and comparative evaluation, ACM Trans. Sen. Netw. 10 (1) (2013) 5:1–5:44. doi:10.1145/2528948. URL http://doi.acm.org/10.1145/2528948

A. P. Miettinen, J. K. Nurminen, Energy efficiency of mobile clients in cloud computing, in: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'10, USENIX Association, Berkeley, CA, USA, 2010, pp. 4–4. URL http://dl.acm.org/citation.cfm?id=1863103.1863107

[4] H. Li, K. Ota, M. Dong, Learning iot in edge: Deep learning for the internet of things with edge computing, IEEE Network 32 (1) (2018) 96–101. doi: 10.1109/MNET.2018.1700202.

[5] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, D. S. Nikolopoulos, Challenges and opportunities in edge computing, in: 2016 IEEE International Conference on Smart Cloud (SmartCloud), 2016, pp. 20–26. doi:10.1109/SmartCloud.2016.18.

[6] D. Laiymani, A. Makhoul, Adaptive data collection approach for periodic sensor networks, in: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2013.

[7] C. Habib, A. Makhoul, R. Darazi, R. Couturier, Real-time sampling rate adaptation based on continuous risk level evaluation in wireless body sensor networks, in: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2017.

[8] G. B. Tayeh, A. Makhoul, D. Laiymani, J. Demerjian, A distributed realtime data prediction and adaptive sensing approach for wireless sensor networks, Pervasive and Mobile Computing 49 (2018) 62 – 75.

[9] H. Harb, A. Makhoul, C. A. Jaoude, En-route data filtering technique for maximizing wireless sensor network lifetime, in: 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), 2018, pp. 298–303. doi:10.1109/IWCMC.2018.8450348.

[10] V. Alieksieiev, One approach of approximation for incoming data stream in iot based monitoring system, in: 2018 IEEE Second International Conference on Data Stream Mining Processing (DSMP), 2018, pp. 94–97. doi:10.1109/DSMP.2018.8478466.

[11] J. Azar, A. Makhoul, R. Darazi, J. Demerjian, R. Couturier, On the performance of resource-aware compression techniques for vital signs data in wireless body sensor networks, in: 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM), 2018, pp. 1–6. doi: 10.1109/MENACOMM.2018.8371032.

[12] J. Azar, R. Darazi, C. Habib, A. Makhoul, J. Demerjian, Using dwt lifting scheme for lossless data compression in wireless body sensor networks, in: 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), 2018, pp. 1465–1470. doi:10.1109/IWCMC.2018. 8450459.

[13] Kshirsagar, P. R., Reddy, D. H., Dhingra, M., Dhabliya, D., & Gupta, A. (2022a). A Review on Comparative study of 4G, 5G and 6G Networks. 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), 1830–1833. IEEE.

[14] H. Harb, A. Makhoul, C. A. Jaoude, A real-time massive data processing technique for densely distributed sensor networks, IEEE Access 6 (2018) 56551–56561.

[15] M. Gaeta, V. Loia, S. Tomasiello, Multisignal 1-d compression by ftransform for wireless sensor networks applications, Appl. Soft Comput. 30 (C) (2015) 329–340. doi:10.1016/j.asoc.2014.11.061. URL http://dx.doi.org/10.1016/j.asoc.2014.11.061

[16] L. Cheng, S. Guo, Y. Wang, Y. Yang, Lifting wavelet compression based data aggregation in big data wireless sensor networks, in: 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), 2016, pp. 561–568. doi:10.1109/ICPADS.2016.0080.

[17] Fragkiadakis, P. Charalampidis, E. Tragos, Adaptive compressive sensing for energy efficient smart objects in iot applications, in: 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE), 2014, pp. 1–5. doi:10.1109/VITAE.2014.6934488.

[18] S. Di, F. Cappello, Fast error-bounded lossyhpc data compression with sz, in: 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Vol. 00, 2016, pp. 730–739. doi:10.1109/IPDPS.2016.11. URL doi.ieeecomputersociety.org/10.1109/IPDPS.2016.11

[19] J. A. Healey, R. W. Picard, Detecting stress during real-world driving tasks using physiological sensors, IEEE Transactions on Intelligent Transportation Systems 6 (2) (2005) 156–166. doi:10.1109/TITS.2005.848368.

[20] A. L. G. et al., Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals, Circulation.

[21] S. Ollander, Wearable sensor data fusion for human stress estimation, Master's thesis, Technical University of Linkoping University (2015).

[22] A. Greco, G. Valenza, A. Lanata, E. P. Scilingo, L. Citi, cvxeda: A convex optimization approach to electrodermal activity processing, IEEE Transactions on Biomedical Engineering 63 (4) (2016) 797–804. doi: 10.1109/TBME.2015.2474131.

[23] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, USA, 2010, pp. 807–814. URL http://dl.acm.org/citation.cfm?id=3104322.3104425

[24] Chaudhury, S., Dhabliya, D., Madan, S., & Chakrabarti, S. (2023). Blockchain Technology: A Global Provider of Digital Technology and Services. In Building Secure Business Models Through Blockchain Technology: Tactics, Methods, Limitations, and Performance (pp. 168–193). IGI Global.

[25] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980. arXiv:1412.6980. URL http://arxiv.org/abs/1412.6980

[26] A. Deligiannakis, Y. Kotidis, N. Roussopoulos, Compressing historical information in sensor networks, in: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD ’04, ACM, New York, NY, USA, 2004, pp. 527–538. doi:10.1145/1007568.1007628. URL http://doi.acm.org/10.1145/1007568.1007628

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958. URL http://jmlr.org/papers/v15/srivastava14a.html

[28] Veeraiah, V., Pankajam, A., Vashishtha, E., Dhabliya, D., Karthikeyan, P., & Chandan, R. R. (2022). Efficient COVID-19 Identification Using Deep Learning for IoT. 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), 128–133. IEEE.