JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 5 No. 2 (2024) 151-160
https://publisher.uthm.edu.my/ojs/index.php/jscdm

LSTM Autoencoders for Internet of Things Data Compression and Battery Conservation

Hussain Falih Mahdi[1]*, Tanupriya Choudhury[2], Yousif Khalid Yousif[3], Mohammed Al Jameel[4], Walid Abushiba[5]

[1] *Computer Engineering Department, College of Engineering, University of Diyala, Baqubah, IRAQ*
[2] *School of Computer Science, University of Petroleum and Energy Studies UPES, Dehradun, Uttarakhand, INDIA*
[3] *Technical Engineering, College for Computer and AI, Northern Technical University, 41000 Mosul, Nineveh, IRAQ*
[4] *Department of Computer Techniques Engineering, College of Engineering, Al-Mustaqbal University, 51001 Hillah, Babylon, IRAQ*
[5] *College of Engineering, Applied Science University, P.O.Box 5055, Manama, KINGDOM OF BAHRAIN*

*Corresponding Author: dr.hussain.mahdi@uodiyala.edu.iq
DOI: https://doi.org/10.30880/jscdm.2024.05.02.011

Article Info

Received: 18 May 2024
Accepted: 10 October 2024
Available online: 18 December 2024

Keywords

Internet of Things, wireless communication, long short term memory (LSTM), autoencoders, Human Activity Recognition

Abstract

This research focuses on a novel data compression technique in an Internet of Things (IoT) based digital communication system. The work simulated a wireless system being transmitted over a Rayleigh fading channel using Phase Shift Keying (M-PSK). The transmitted data has been sourced from Human Activity Recognition (HAR) using wearable devices and an open-source application. Unlike conventional compression techniques, this study uses Long Short-Term Memory (LSTM) Autoencoders to transform data in low-dimensional form in the wireless system. The main purpose of this study is to evaluate the efficiency of LSTM Autoencoders by reducing data dimensionality while maintaining important information for better activity recognition accuracy. This approach leads to better conservation in IoT devices. The proposed method's performance has been evaluated at various compression levels and modulation levels. The Bit Error Rate (BER) vs. Signal Noise Ratio (SNR) curves of the M-PSK system have been evaluated in comparison with the Mean Square Error (MSE) of the compression and decompression. A trade-off between compression ratio and MSE has been illustrated, which ultimately leads to determining the accuracy of the measurement of human activity. The results highlight the benefits of leveraging the power of LSTM Autoencoders for data compression in the communication of the wireless system. The results contribute to advancing wearable HAR systems and this general performance optimization of the IoT system.

1. Introduction

In recent times, Internet of Things (IoT) and wearable devices have witnessed exponential growth where data is collected and transmitted for various applications, and using various transducers like accelerometers, gyroscopes, and motion sensors, etc. [1]. This has led to the introduction of a wealth of opportunities, including HAR, which is gaining traction in applications that include personalized healthcare monitoring, behavioral analysis, and contextaware services [2]. Nevertheless, the massive data produced from these devices poses many challenges in the context of suboptimal transmissions, storage limitations, and real-time processing of the data [3]. This issue leads to a need for robust and efficient data compression methods that ensure seamless and reliable data transmissions. Some of the conventionally used data compression methods involve Huffman Coding and Lempel-Ziv-Welch (LZW), which have the ability to compress data to a certain extent and thus reduce the transmission load [4]. Yet, the raw sensor data exhibit a wider degree of randomness that makes the performance of these compression methods to be less optimal. The inherently sequential nature of the wearable sensors exhibits wider temporal dependencies and correlations that conventional compression methods may fail to capture. As a result, a loss of critical information may occur, or the compression may keep the wireless channel busy transmitting a large amount of information [4], [5].

Wireless sensing devices transmit information over a variety of wireless networks that exhibit real-world fading and noise characteristics. Additionally, modulation techniques can vary, including the use of phase shift keying, amplitude modulation, and orthogonal frequency division multiplexing (OFDM). These modulation techniques are immensely used in data transmission due to their ability to combine multiple data bits into symbols, thus reducing the bandwidth requirements. Combining these modulation methods with advanced compression techniques enables the system designers to enhance the data transmission rates while also maintaining data integrity and critical information. Yet, the inclusion of data compression techniques remains critical to preserve the network resources and to conserve the energy consumption of the IoT and wearable devices, themselves.

To conserve the transmission resources and to address the limitation associated with conventional data compression techniques, the use of LSTM Autoencoders [6], [7] has been proposed as part of this work. By harnessing the temporal dependencies contained in the data, the model has the ability to capture intricate patterns that lead to data compression. The LSTM-Autoencoders transform the data into a low-dimensional latent space, leaving room for reconstruction. By the employment of LSTM Autoencoders, the study aims to present a data compression method, where the transmission will be made over a wireless channel employing M-PSK modulation. The real-world representation of the communication system has been simulated by introducing a Rayleigh fading channel and Gaussian noise. Modulation has been introduced at numerous levels by M-PSK. The ability of LSTM Autoencoders for data compression and their immunity towards the impairments brought under the wireless communication channels has been investigated under varying SNR conditions. By simulating various levels of information and systems, the performance analysis focuses on evaluating trade-offs between the data compression and reliable data transmissions of multi-dimensional HAR data.

This study aims to design and implement a deep learning-based framework using LSTM Autoencoders. The design is based on transmitting HAR data from wearable gears over channels that communicate wirelessly under the simulation of the M-PSK modulation model and the Rayleigh channel. In addition, the investigation of the benefits that LSTM Autoencoders offer as compared to the traditional compression techniques, is also performed. The system efficiency was investigated as a trade-off between the level of compression and the data rate as well as reconstructed data of MSE.

2. Literature Review

Energy conservation in IoT devices is critical in prolonging battery life and ensuring seamless and efficient operations. Several methods have been proposed in research. [8]–[10] presents a low-power model-based approach to IoT battery life conservation in the form of increased sleep and standby time. In this approach, non-essential components are turned off, and the device keeps running under low power requirements until the important events take place. Another approach presented in research is power duty cycling [11], [12]. In this method, the device is made to alternate between sleep periods and active periods to reduce the overall power consumption. The third approach towards energy conservation is the adoption of efficient communication protocols [13], [14]. Some of the advanced technologies that are adopted under this approach involve Zigbee, Bluetooth Low Energy (BLE), etc. Energy harvesting serves as another method where energy production devices like solar panels or kinetic energy generators are attached to the IoT devices, as discussed in [15]–[18]. The fifth and one of the potential techniques in battery conversation of IoT devices is through data compression and aggregation. This approach employs numerous techniques where data from various devices is gathered and processed in a single go, thus reducing the frequency of data transmission requirements. The data compression further transforms the higher dimensionality of data into lower dimensions that allow such devices to operate in low-power states for longer periods.

Data compression and IoT batter conservation have remained active topics in the research. With the advent of wider connected networks and wearable devices, researchers are actively finding ways to address the challenges associated with higher levels of data generation and explore reliable data compression methods. In [4], the survey has been presented on various data compression techniques that are adopted based on the application (data type), data quality, and coding approaches. The approaches that have been targeted involve Huffman coding, Run Length Encoding (RLE), predictive transform coding techniques, dictionary-based coding, fractional compression, Burrows-Wheeler Transform (BWT), and quantization. These generally fall in the domain of conventional approaches and, therefore, have limited applicability to the data emerging from IoT devices. Some of the similar surveys have been conducted in [19]–[22].

The use of machine learning and deep learning is increasingly making its way into data compression. One such commonly employed architecture is the use of Autoencoders that use neural networks to transform input data into a compact representation. Residing on a continuous evaluation of the encoding and decoding process, Autoencoders identify those compression features that can reduce the loss of information while representing data in a latent space. By minimizing reconstruction errors, autoencoders retain critical information to the maximum possible extent. Numerous architectures are employed in the context, including feedforward Autoencoders where fully connected layers perform the encoding and decoding operations [23]–[25]. Another type of autoencoder employed is the convolutional autoencoder, which uses convolutional layers. Such Autoencoders are found to be effective in capturing the spatial information and patterns contained in the data, which aid in reducing the dimensionality [26], [27]. One of the recent forms of Autoencoders is the recurrent Autoencoders that are used to analyze time series sequences of data. These Autoencoders generally fall in the domain of deep Autoencoders, where long- and short-term memory is employed. Details on the recurrent Autoencoders have been provided in the next section of this paper.

The current applications for the use of Autoencoders for data compression in IoT devices in wireless networks are limited and only a few studies have investigated this method. One such closest study has been found in [28], which performs EEG data compression using the convolutional Autoencoders. This study is related to the present case since EEG data exhibit similar random patterns which are generally found in the data emanating from IoT and wearable devices. The method makes use of deep convolutional Autoencoders comprising 27 layers, thus transforming signals into low-dimensional variants. The deep learning approach allows the model to learn representations of both low and high levels of signals, enabling minimal loss during signal reconstruction. The proposed CAE model was evaluated on a dataset of 4800 ECG fragments from 48 patients, achieving a compression rate of 32.25 and an average Percentage Root Mean Squared Difference (PRD) value of 2.73%. These promising results suggest that the deep model can securely transfer data in a low-dimensional form to remote medical centers, making it suitable for applications in wearable devices, e-health, telemetry, and Holter systems.

3. Methodology

Developing the deep learning Autoencoders model for data compression in wireless communication systems has involved various steps, including downloading data, preprocessing, architecture development, and wireless channel simulations. This section presents each of these stages.

3.1 Deep Autoencoders

The use of Deep LSTM Autoencoders was to benefit from LSTM gated architecture as well as unconfirmed approaches in order to reduce the dimension of the dataset. The input data, which is highly dimensional, consists of two models: the encoder and the decoder. This data is converted into a hidden space depiction by means of the encoder and is sent back to the originated form using the decoder. By capturing long-term dependencies existing during the time-based patterns of the sequential data, the encoder's architecture enables it to offer a suitable illustration of the data. Due to this, they are specifically well-suited to manage HAR data with an elevated level of unpredictability. Several LSTM layers make up the encoder architecture, which processes and transforms sequential data and updates according to the hidden states. The data is compressed into a latent space by the last layer of the LSTM architecture. The decoder part comprises LSTM or dense layers that take the compressed representation of the input data and transform them back into their original shape and dimensions. The process of mapping and reverse mapping is controlled using the loss function, whose primary aim is to minimize the reconstruction error and, thus, the discrepancies between the original and reconstructed data. The most commonly employed function is the MSE, which measures the average squared difference between the input data and its reconstructed variants. The training of the Autoencoders involves an unsupervised process where high-dimensional data is transformed into a lower-dimensional latent space. Optimization algorithms, including the Stochastic Gradient Descent (SDG) or Adam, are backed by this.

Numerous states govern the LSTM Autoencoders model. Fundamentally, there are three gates involved: forget gates, input gates, and output gates. These gates are governed by hidden states h that lead to cell state c during the given time step t.

$$f_t = \sigma \left(W_f \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma \left(W_i \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$g_t = \tanh\left(W_c\,[h_{t-1}, x_t] + b_c\right) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{4}$$

$$o_t = \sigma\left(W_o\,[h_{t-1}, x_t] + b_o\right) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

Where f_t represents the forget gate, i_t represents the input gate, o_t represents the output gate, g_t represents the candidate cell state, c_t represents the updated cell state, and h_t represents the hidden state. W and b notations are the weighted matrices and bias vectors, respectively, of the individual gates. $\odot$ is the Hadamard product, and σ represents the sigmoid activation function.

The performance metric MSE is administered through a squared difference between the original data samples x_i and reconstructed data samples $x_i{}'$. Mathematically, it is administered by the following equation:

$$MSE(x, x') = \frac{1}{N}\sum_i (x_i - x_i')^2 \tag{7}$$

Figure 1 provides the general architecture of the LSTM-based AutoencoderS for data compression. The encoder aims to transform high-dimensional data into a low-dimensional latent space, which is then reconstructed by the decoder. Equations 1 through 7 presented above administer the overall functionality of the model.

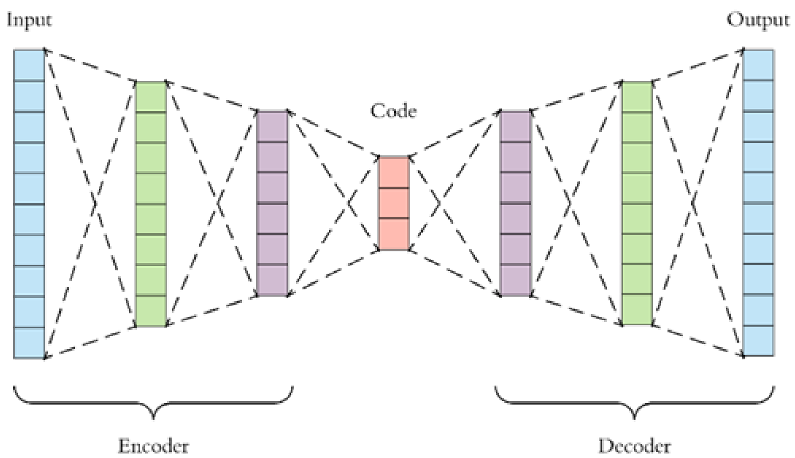

Fig. 1 *General LSTM Autoencoders architecture*

3.2 LSTM Autoencoders Architecture

The beginning of the construction of the initial form of the model involves normalizing this data in the standardized scales. This was done for every data point by subtracting the mean and then dividing it by SD, which ensured that the value had a unit difference and a value around zero. This step assists in enhancing this model's effectiveness by using the normalization process.

The subsequent procedure to normalization is data pre-processing, which is done to feed the data to the necessities of LSTM layers. The encoder part of the model includes several LSTM layers, as the HAR data has a timestamp and a serial dependence property. As for the encoder, two LSTM layers with 64 and 32 units are accurately included. Over time, these two layers have been responsible for identifying the complex relationship and dependency in data. The rich data that is highly dimensional was compressed and made into less dimensional data through the use of a dense layer. This layer is helpful in accurately and easily calculating only the necessary attributes required for the model's strength. Full connection layers are incorporated within the decoder part to reconstruct the data. The first layer of the decoder combines 32 units and progressively reconstructs the compressed data rising from it. From the second fully connected layer, the data is forwarded into the expected input shape in order to ensure a very good reconstruction of the data fed to the AI model. These layers use linear activation functions and ReLU (Rectified Linear Unit) activation functions, which help with the reconstruction process in retaining data stability plus the inclusion of non-linearity.

The autoencoder model is obtained by cascading two models, which are the encoder and decoder models, after the models have been built. This integrated model effectively informs the encoder block and discharge output at the decoder segment for compressing and reconstructing the given data. Moreover, the encoder and decoder components process the data autonomously, allowing for some flexibility in data-related pipelining tasks. Compared to the initial dimension 561, the encoding dimensions were ranging from 32 to 300 while training the model with the normalized data set. Another advantage is that such diversity of encoding dimensions makes searching for the best applicable balance between compression and the degree of the reconstructed item's similarity to the original simpler. Training is a process that adopts procedures such as the learning rate, batch size, and number of epochs to get the best results.

Table 1 *Training parameters of the LSTM Autoencoders model*

Parameters	Values
And an entry	Samples, 561
Input Data Dimensions	32 to 300
Encoded Data Dimensions	50
Epochs	80:20
Data Split	32
Batch Size	Adam
Compiler	MSE
Loss Function	Samples, 561

The model architecture developed in this study has been presented in Figure 2. Having transformed the high dimensional data into encoded form at varying levels, the next stage is to transmit and receive the data over a wireless communication channel. The fundamental idea is to analyze the immunity of the Autoencoders model at varying SNR conditions. For instance, deviations in the data can be analyzed when the data propagates through a Rayleigh fading channel, further distorting the reconstruction process. Additionally, the models have been designed to analyze the increase or decrease in the data rates at varying levels of compression and modulation. The wireless system design transforms the encoded data into a binary representation, which is then modulated to make symbols. These symbols are passed through a Rayleigh fading channel where Gaussian noise is also added to the model. Thus, the SNR conditions start varying. The demodulation process is adopted at the receiver side, where encoded data is received. This data is transformed back into the original representation, and the corresponding error is measured.

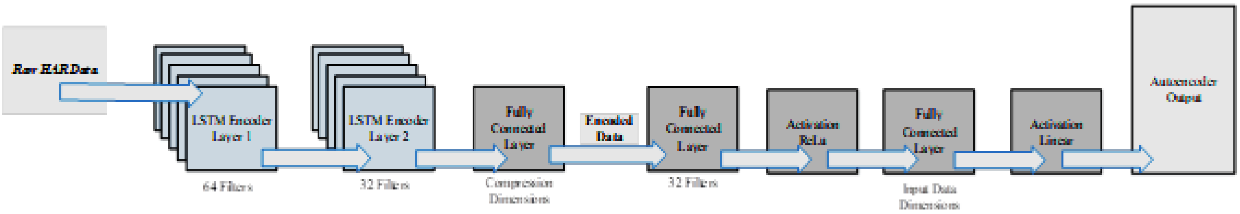

Fig. 2 *Proposed LSTM Autoencoders architecture*

3.3 Dataset

The dataset involves Human Activity Recognition (HAR) data recorded using a smartphone's gyroscope and accelerometer sensors. This data is available as open source and was fundamentally gathered at the Smartlab – Non-Linear Complex Systems Laboratory in Italy. [29]. To collect the data, experiments were conducted on a group of 30 individuals aged between 19 and 48 years. Each participant performed six different activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying. While this was happening, the participants carried a smartphone on their waist mimicking wearable gear, in order to ensure that these collected data could represent real-world usage. The gyroscopic and accelerometer data were captured at a sampling rate of 50 Hz as 3-axial acceleration and angular velocity. A well-detailed motion pattern can be collected by using high-frequency data collection, and this step is very important for the accuracy of the activity to be recognized. Labeling every activity in the dataset was performed despite the un-usage of these labels for compression of the data. 70% of the data was used for training, and 30% was used for testing, which ensures a reliable assessment of the model's functionality. The data was pre-processed, and this step includes several stages for enhancing the data quality and the data's fitness for investigation. To smooth out any abnormalities and lessen the impact of transient noise, a filter was used to reduce the windowed data parts by 2.56 seconds with a 50% overlay. A low-pass Butterworth filter was used for additional enhancing the data, especially for the segregated body motion and gravitational elements. The effective separation of the higher-frequency body motion signals and the low-frequency gravitational force was achieved by using this filter. Additionally, the lower frequency gravitational force was discarded using a high-pass filter with a 0.3 Hz cutoff frequency to ensure that only the applicable motion data is engaged. Each of the thousands of samples in the resultant dataset has 561 features that replicate angular velocity and triaxial acceleration. This wide-ranging and ironic dataset provides a thorough basis for generating and assessing machine learning models for human activity recognition. By utilizing this data, scholars can gain a significant understanding of the trends and attributes of numerous activities, eventually progressing with wearable technology and activity-tracking systems.

3.4 Performance Metrics

The system's performance has been determined based on three parameters. First, the system's loss performance has been analyzed using MSE. This determined the model's training accuracy in encoding and reconstructing the data. The second measure of accuracy is to determine the improved bandwidth consumption by transmitting the same amount of data in less time due to compression. This has been analyzed using various levels of compression and modulation. At this stage, the immunity of the deep learning model towards varying SNR conditions has also been analyzed. Finally, the battery conversation of the IoT devices has been discussed, considering the lower battery consumption when transmitting compressed data compared to the original high-dimensional data. These performance metrics are presented in the next section of this paper.

4. Results

In this section, the performance outcomes of the model at various stages have been discussed and presented.

4.1 LSTM Autoencoders Training Performance

The LSTM model has been developed and trained using a training and validation dataset (the testing dataset has been used as validation in this case). This ensures that the model has been accurately trained without undergoing overfitting or under-fitting, and the mean square error has been determined against the validation dataset. The outcomes against different compression ratios have been depicted in Figure 3. It can be seen that for higher compression rations. It can be seen that by using a deep LSTM Autoencoders model, the higher data compression ratios exhibit MSE patterns, which are comparable to lower compression ratios. The potential reason is the ability of the model to infer sequential patterns contained in the data rather than focusing on the individual bits. Thus, the trends contained in the samples of information can aid in attaining higher compression ratios while exhibiting comparable MSE performance. Therefore, a system designer can choose higher compression ratios to the point where MSE performance remains optimal and comparable.

To analyze the effectiveness of the model, the Percentage RMS Difference Normalized (PRDN) values have been analyzed compared to the architecture developed in [28]. It has been found that by using a relatively simplified architecture (2 LSTM layers), the model still outperforms the one presented in [28], where a highly convoluted architecture has been developed comprising 27 CNN layers. The average PRDN value for various compression levels is 0.0079 percent compared to a value of 31.17 percent in [28].

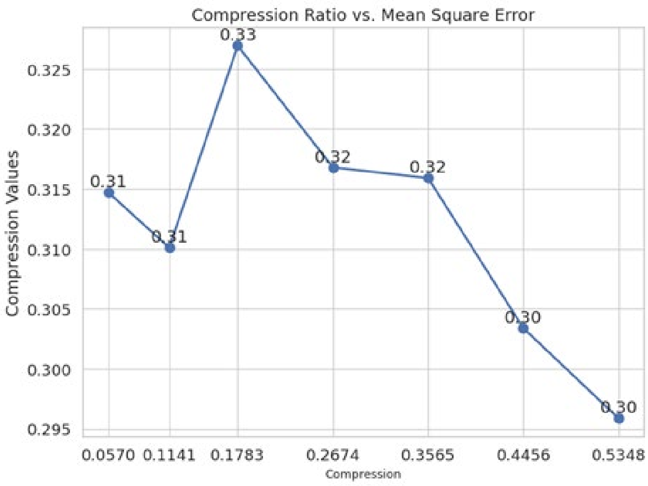

Fig. 3 *Mean square error against different compression ratios*

4.2 BER and SNR Performances

To ensure that the communication system exhibits the desired response under the Rayleigh fading a QPSK, the BER vs SNR curves have been plotted and are found to be comparable with [30], [31]. The results are depicted in Figure 4. The immunity of the Autoencoders' MSE performance with the SNR variations has been analyzed in addition. It has been found that the decoder exhibits the same MSE value regardless of the SNR value because of its innate nature of residing on the data trends rather than focusing on the specific values. The standard deviation between the MSE values of the Autoencoders under various SNR values is minimal. By varying the SNR between -10 dB to +10 dB, it has been found that the values recorded are similar to those presented in Figure 4, against the given compression levels.

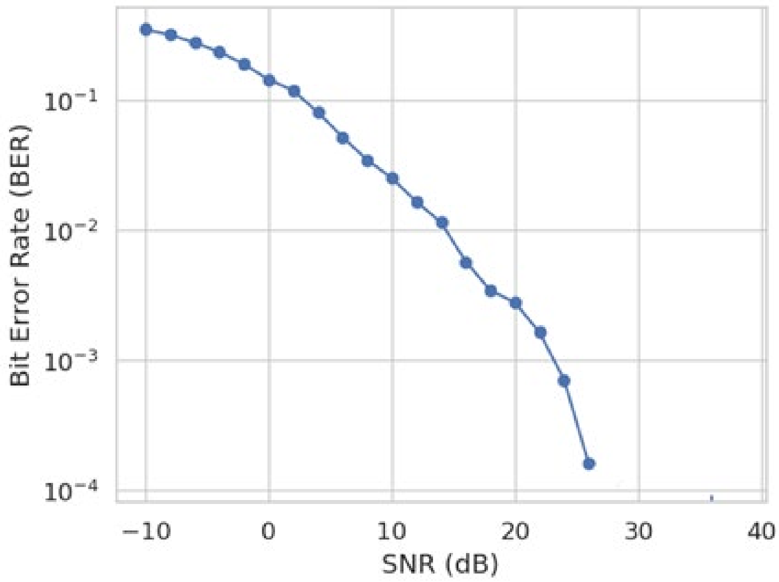

Fig. 4 *4-PSK BER vs. SNR curve*

4.3 Energy Consumption

The energy consumed per sample by an IoT device has been determined from [32] to determine the energy consumption under various compression and modulation levels. Different batteries against groups have been listed while highlighting the power performance of different parts of a wearable device. In Table 2, energy consumed by some of the parts of a wearable device has been presented.

Table 2 *Training parameters of the LSTM Autoencoders model*

Block	Effective current per unit sample (NA)
Sensor	0.75
Amplifier	1
10-bit SAR ADC	0.61
Transmitter (CC2500)	1.81

Since the study is focusing on transmission power, fewer transmissions can lead to less power consumption. Therefore, by using higher levels of modulation and large compression ratios, the energy consumption is reduced, and the battery life of an IoT device is conserved. This aspect is presented in Figure 5. Thus, by using LSTM Autoencoders-based data compression and a middle / higher order modulation, the system's energy performance can significantly improve, and the battery life of the IoT devices can be conserved.

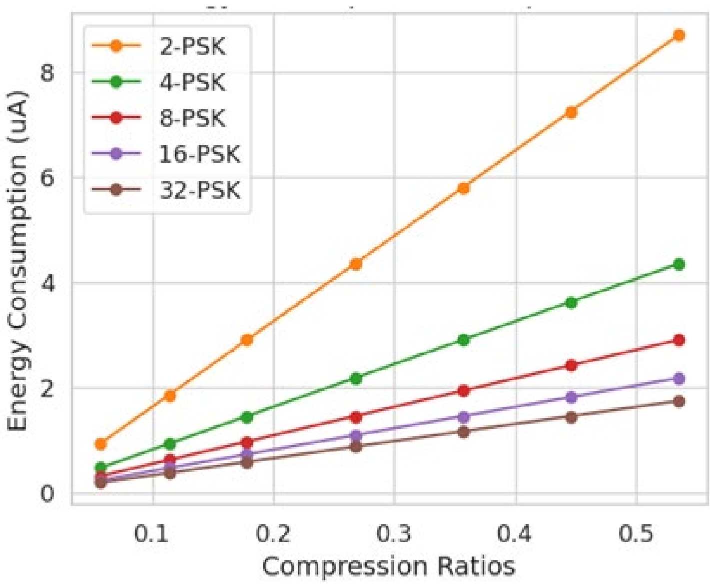

Fig. 5 *Energy consumption vs. compression ratios*

Comparing this method to similar studies reveals several key differences and improvements. The described method reduces energy consumption by optimizing data compression and modulation levels. Similar studies [33] also aim to enhance energy efficiency but may not simultaneously address modulation levels. By combining LSTM Autoencoders-based data compression with higher-order modulation, this method achieves a more comprehensive approach to energy conservation. Other research is devoted to preventing data distortion and achieving very high compression rates. However, they may not particularly ponder the impacts of the modulation levels on energy consumption [34]. A further level of energy saving is provided by the modulation optimization, which is included in the current method and, in contrast, is absent in other similar studies focused on compression only. Moreover, the usefulness of the method is illustrated by applying it to wearable IoT devices. Research might ignore the broader picture of wearable devices in order to examine specific modulation techniques [35]. An area that must have energy efficiency due to short battery life is wearable IoT devices, which are targeted in this study.

5. Conclusion

In this study, we proposed an innovative deep Long Short-Term Memory (LSTM) Autoencoders model for efficient data compression in wearables for HAR. The analysis of the general performance of the model showed that temporal dependencies are preserved, and high compression coefficients are achieved. The LSTM Autoencoders training result showed the model's potential to accurately identify sequential patterns in the Human Activity Recognition (HAR) data. Proposing a deep LSTM Autoencoders design, we observed the conception of the Mean Squared Error (MSE) that does not significantly depend on the chosen data compression ratios, even greater than the ratios used in our study. The model's focus on analyzing sequential data allowed to achieve higher compression ratios without losing MSE efficiency. Because of its adaptability, system designers can select the best compression ratios for their particular needs. Additionally, we contrasted the performance of a highly complex design from [28] with our comparatively simpler LSTM Autoencoders architecture. In terms of PRDN, the model fared better than the 27-layered convolutional architecture by LSTM layers. We also looked at how resistant the Autoencoders were to changes in MSE as SNR changed. Because the decoder focused on data trends rather than particular values, it displayed consistent MSE values across a range of SNR levels. This characteristic ensures robust performance under varying SNR conditions. Finally, we analyzed the energy consumption of IoT devices, particularly wearable devices, under various compression and modulation levels. Higher modulation levels and larger compression ratios reduce energy consumption, thereby preserving the battery life of the IoT devices. Our LSTM Autoencoders-based data compression combined with middle/higher-order modulation significantly improved the system's energy performance and conserved battery life.

Acknowledgment

The authors would like to thank the University of Diyala for supporting this work.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the paper's publication.

Author Contribution

The authors confirm their contributions to the paper as follows: ***study conception and design:*** *Hussain Falih Mahd and Yousif Khalid Yousif;* ***data collection:*** *Hussain Falih Mahdi and Mohammed Al Jameel* ***analysis and interpretation of results:*** *Hussain Falih Mahdi and Walid Abushiba;* ***draft manuscript preparation:*** *Hussain Falih Mahdi, and Tanupriya Choudhury. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Li, S., Song, H., & Iqbal, M. (2019). Privacy and security for resource-constrained IoT devices and networks: research challenges and opportunities. *Sensors*, 19(8), 1935-1939. https://doi.org/10.3390/s19081935

[2] Shah, R., & Chircu, A. (2018). IoT and AI in healthcare: A systematic literature review*. Issues in Information Systems*, 19(3), 33-41. https://doi.org/10.48009/3_iis_2018_33-41

[3] Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud computing issues, challenges and opportunities: A review. *Qubahan Academic Journal*, 1(2), 1–7. https://doi.org/10.48161/qaj.v1n2a36

[4] Jayasankar, U., Thirumal, V., & Ponnurangam, D. (2021). 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*, 33(2), 119–140. https://doi.org/10.1016/j.jksuci.2018.05.006

[5] Gao, Z., Dai, L., Han, S., Chih-Lin, I., Wang, Z., & Hanzo, L. (2018). Compressive sensing techniques for next-generation wireless communications. *IEEE Wireless Communications*, 25(3), 144–153. https://doi.org/10.1109/MWC.2017.1700147

[6] Dasan, E. & Panneerselvam, I. (2021). A novel dimensionality reduction approach for ECG signal via convolutional denoising autoencoder with LSTM. *Biomedical Signal Processing and Control*, 63, 102225. https://doi.org/10.1016/j.bspc.2020.102225

[7] Pawar, K., & Attar, V.Z. (2020). Assessment of Autoencoder Architectures for Data Representation. In Pedrycz, W., Chen, SM. (eds), *Deep Learning: Concepts and Architectures. Studies in Computational Intelligence*, Springer, Cham. https://doi.org/10.1007/978-3-030-31756-0_4

[8] Pasricha, S., Ayoub, R., Kishinevsky, M., Mandal, S. K., & Ogras, U. Y. (2020). A survey on energy management for mobile and IoT devices. *IEEE Design & Test*, 37(5), 7–24. https://doi.org/10.1109/MDAT.2020.2976669

[9] Callebaut, G., Leenders, G., Van Mulders, J., Ottoy, G., De Strycker, L., & Van der Perre, L. (2021). The art of designing remote iot devices—technologies and strategies for a long battery life. *Sensors*, 21(3), 913-940. https://doi.org/10.3390/s21030913

[10] Froytlog, A. et al., (2019). Ultra-low power wake-up radio for 5G IoT. *IEEE Communications Magazine*, 57(3), 111–117. https://doi.org/10.1109/MCOM.2019.1701288

[11] Rana, B., &Singh, Y. (2022). Duty-Cycling Techniques in IoT: Energy-Efficiency Perspective. In: Singh, P.K., Singh, Y., Kolekar, M.H., Kar, A.K., Gonçalves, P.J.S. (eds), *Recent Innovations in Computing. Lecture Notes in Electrical Engineering*, Springer. https://doi.org/10.1007/978-981-16-8248-3_42

[12] Monti, A., Alata, E., ragomirescu, D. D., & Takacs, A. (2018). Power supply duty cycling for highly constrained IoT devices. *2018 International Semiconductor Conference (CAS)*, IEEE, 215–218. https://doi.org/10.1109/SMICND.2018.8539832

[13] Bello, O., & Zeadally, S. (2019). Toward efficient smartification of the Internet of Things (IoT) services. *Future Generation Computer Systems*, 92, 663–673. https://doi.org/10.1016/j.future.2017.09.083

[14] Behera, T. M., Samal, U. C., & Mohapatra, S. K. (2018). Energy-efficient modified LEACH protocol for IoT application. *IET Wireless Sensor Systems*, 8(5), 223–228. https://doi.org/10.1049/iet-wss.2017.0099

[15] Zeadally, S., Shaikh, F. K., Talpur, A., & Sheng, Q. Z. (2020). Design architectures for energy harvesting in the Internet of Things. *Renewable and Sustainable Energy Reviews*, 128, 109901. https://doi.org/10.1016/j.rser.2020.109901

[16] Adila, A. S., Husam, A., & Husi, G. (2018). Towards the self-powered Internet of Things (IoT) by energy harvesting: Trends and technologies for green IoT. *2018 2nd international symposium on small-scale intelligent manufacturing systems (SIMS)*, IEEE, 1–5. https://doi.org/10.1109/SIMS.2018.8355305

[17] Elahi, H., Munir, K., Eugeni, M., Atek, S., & Gaudenzi, P. (2020). Energy harvesting towards self-powered IoT devices. *Energies*, 13(21), 5528. https://doi.org/10.3390/en13215528

[18] Yau, C.-W., Kwok, T. T.-O., Lei, C.-U., & Kwok, Y.-K. (2018). Energy harvesting in internet of things. *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, 0, 35–79. https://doi.org/10.1007/978-981-10-5861-5_3

[19] Smith, C. A. (2010). A survey of various data compression techniques. *Int J pf Recent Technol Eng*, 2(1), 1–20.

[20] Hosseini, M. (2012). *A survey of data compression algorithms and their applications*. Network Systems Laboratory, School of Computing Science, Simon Fraser University, BC, Canada.
[21] Sheltami, T., Musaddiq, M., & Shakshuki, E. (2016). Data compression techniques in wireless sensor networks. *Future Generation Computer Systems*, 64, 151–162. https://doi.org/10.1016/j.future.2016.01.015
[22] Rekha, H., & Samundiswary, P. (2015). Survey on low power adaptive image compression techniques for WSN. *2015 International Conference on Communications and Signal Processing (ICCSP)*, 1128–1132. https://doi.org/10.1109/ICCSP.2015.7322679
[23] Bandara, G., & Siyambalapitiya, R. (2020). Deep autoencoder-based image compression using multi-layer perceptrons. *International Journal of Soft Computing and Engineering*, 9(6), 1–6. https://doi.org/10.35940/ijsce.E3357.039620
[24] Watkins, Y., Iaroshenko, O., Sayeh, M., & Kenyon, G. (2018). Image compression: Sparse coding vs. bottleneck autoencoders. *2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pp. 17–20. https://doi.org/10.48550/arXiv.1710.09926
[25] Berezkin, A., Slepnev, A., Kirichek, R., Kukunin, D., & Matveev, D. (2021). Data compression methods based on Neural Networks. *The 5th International Conference on Future Networks & Distributed Systems*, 511–515. https://doi.org/10.1145/3508072.350817
[26] Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2019). Energy compaction-based image compression using convolutional autoencoder. *IEEE Transactions on Multimedia*, 22(4), 860–873. https://doi.org/10.1109/TMM.2019.2938345
[27] Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2018). Deep convolutional autoencoder-based lossy image compression. *2018 Picture Coding Symposium (PCS)*, 253–257. https://doi.org/10.1109/PCS.2018.8456308
[28] Yildirim. O., Tan. R. S., & Acharya, U. R. (2018). An efficient compression of ECG signals using deep convolutional autoencoders. *Cognitive Systems Research*, 52, 198–211. https://doi.org/10.1016/j.cogsys.2018.07.004
[29] Human Activity Recognition Using Smartphones. (Aug. 02, 2023). *Ddataset by uci data world*. https://data.world/uci/human-activity-recognition-using-smartphones
[30] Bala, D., Waliullah, G. M., Islam, N., Abdullah, I., & Hossain, M. A. (2021). Analysis the probability of bit error performance on different digital modulation techniques over AWGN channel using MATLAB. *Journal of Electrical Engineering, Electronics, Control and Computer Science*, 7(3), 9–18.
[31] Wu, D., Nekovee, M., & Wang, Y. (2020). Deep learning-based autoencoder for m-user wireless interference channel physical layer design. *IEEE Access*, 8, 174679–174691. https://doi.org/10.1109/ACCESS.2020.3025597
[32] Caiazza, C., Luconi, V., & Vecchio, A. (2024). Energy consumption of smartphones and IoT devices when using different versions of the HTTP protocol. *Pervasive and Mobile Computing*, 97, 101871. https://doi.org/10.1016/j.pmcj.2023.101871
[33] Azar, J., Tayeh, G.B., Makhoul, A. et al. (2022). Efficient lossy compression for IoT using SZ and reconstruction with 1D U-net. *Mobile Networks and Applications*, 27, 984–996. https://doi.org/10.1007/s11036-022-01918-6
[34] Liu, J., Djukic, P., Kulhandjian, M., & Kantarci, B. (2024). Deep dict: deep learning-based lossy time series compressor for IoT data. *arXiv*, 2401, 10396. https://doi.org/10.48550/arXiv.2401.10396
Gao, H., Lu, Y., Yang, S., Tan, J., Nie, L., & Qu, X. (2024). Energy Consumption Analysis for Continuous Phase Modulation in Smart-Grid Internet of Things of beyond 5G. *Sensors*, 24, 533. https://doi.org/10.3390/s24020533