Organization behavior
Shaked Chen – 207253220

1. **Efficient Data Compression for Neural Recordings**

Neuralink, a company advancing brain-computer interface (BCI) technology, faces a challenge in managing and transmitting neural data. The N1 implant generates approximately 200 megabits per second (Mbps) of data, but its wireless transmission capability is limited to 1 Mbps. This highlights the need for advanced data compression techniques that reduce data size while preserving essential information.

Efficient data compression is critical for real-time processing and transmission of neural signals, particularly for applications like prosthetic control and communication. The challenge lies in achieving high compression ratios with minimal reconstruction error, ensuring compressed data retains its integrity. Algorithms must also be computationally efficient to meet Neuralink's hardware and energy constraints.

Addressing this challenge is essential for Neuralink's mission to create seamless communication between the brain and external devices, unlocking the full potential of BCI technology for healthcare and beyond.

Step 1b: Analyze 3 Stakeholders Involved

1. **Neuralink Engineers:** Responsible for designing and refining data compression algorithms that ensure efficient and accurate transmission within hardware limitations.
2. **Regulatory Authorities:** Oversee compliance with safety and ethical standards, ensuring that Neuralink's solutions meet industry guidelines for secure and reliable medical devices.
3. **End-Users (Patients and Researchers):** Depend on compressed neural data for prosthetic control and research, requiring accuracy and usability to achieve meaningful outcomes.

2. **We have chosen the dataset provided by Neuralink**, which consists of one hour of raw electrode recordings from the motor cortex of a non-human primate performing a task. This dataset, sampled at 20 kHz across 1024 electrodes with 10-bit resolution, produces approximately 200 Mbps of uncompressed data. It is ideal for evaluating data compression methods due to its high resolution and complexity.

Solution Methodology:
To address the challenge, we will use an algorithm-based approach. Various lossless compression algorithms, such as Run Length Encoding, Huffman Coding, and Lempel-Ziv-Welch (LZW), will be tested. Each algorithm will be evaluated for its compression ratio, computational efficiency, and processing speed. Hypothesis testing will be employed to compare their performance and identify the

optimal algorithm. Additionally, we may develop a custom compression method tailored to Neuralink's specific data requirements and hardware constraints.

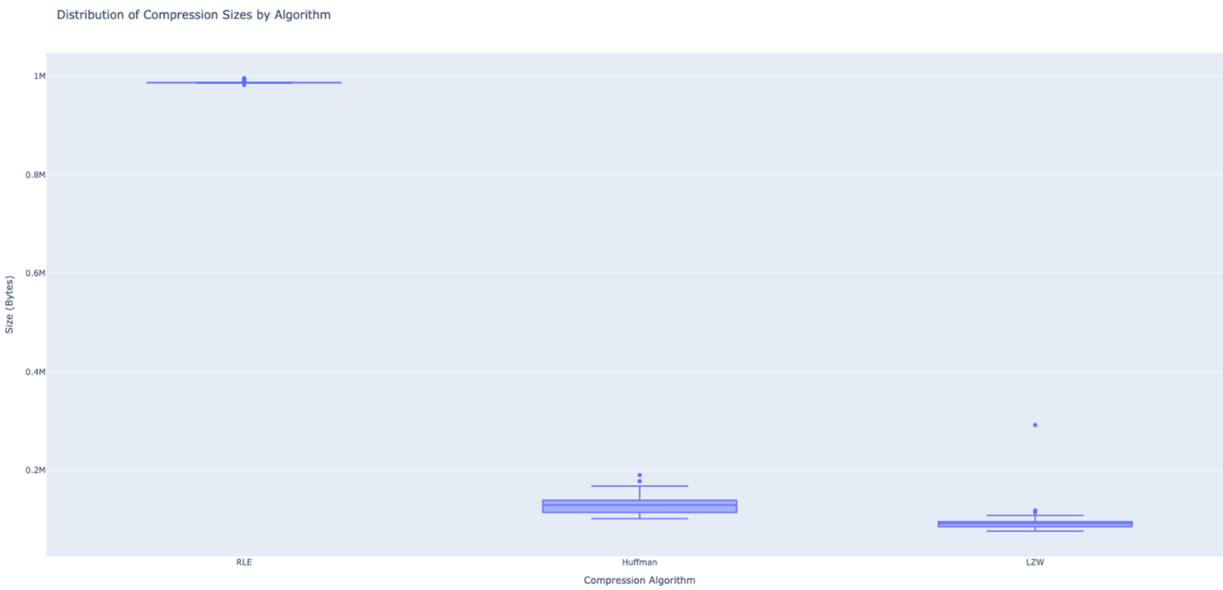
3.

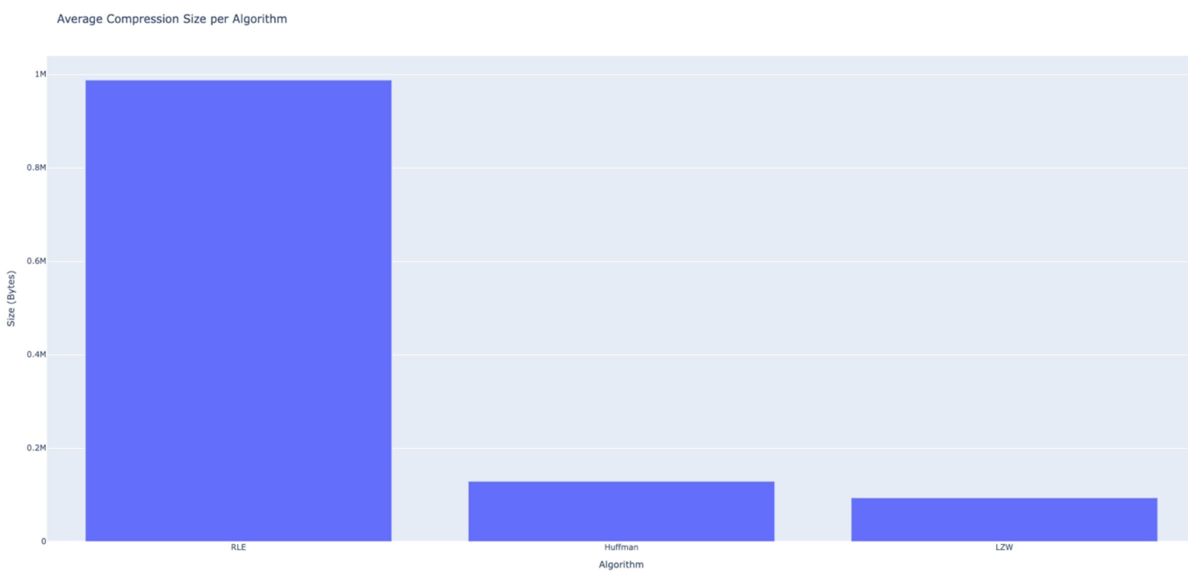
Run Length Encoding source research

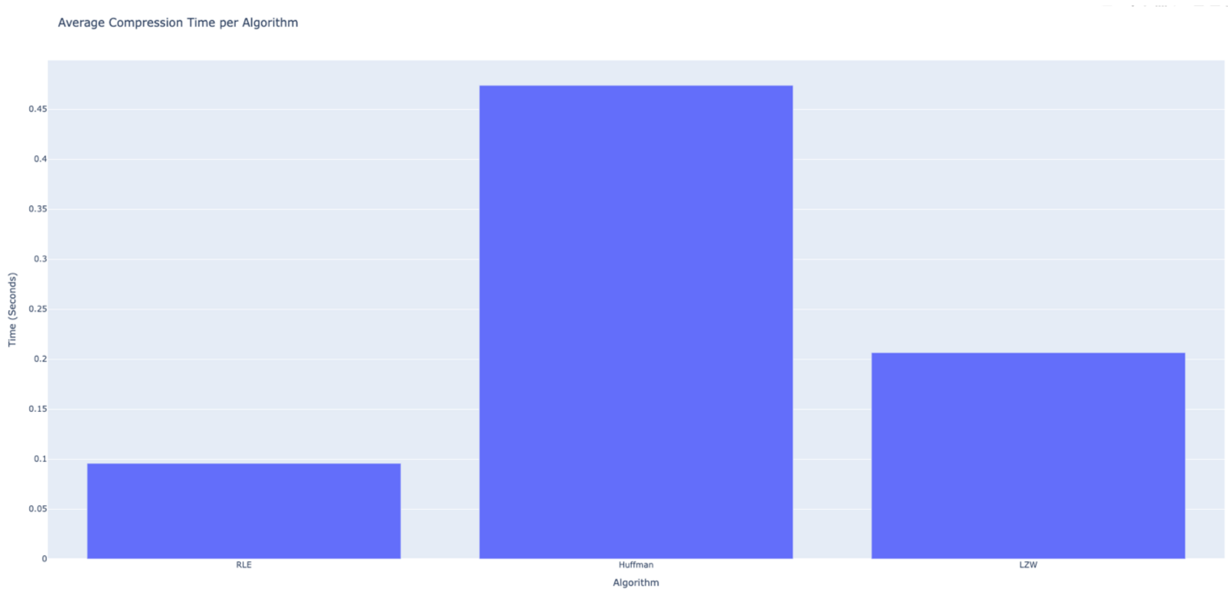
https://d1wqtxts1xzle7.cloudfront.net/49375045/r1-libre.pdf?1475679061=&response-content-disposition=inline%3B+filename%3DAudio_Compression_Based_on_Discrete_Cosi.pdf&Expires=1736515015&Signature=DXvEZ~tb4U1v75AbMStkDSzZyy5Eh9cC40tjPYNLP6bBgy-ynjc3WqZ6xTKBgxa-VpaSoWecUrBfH9-eN0eXdOhHRC-dR8uNP~DGG2OdbS6ibCH-pfewHxMxCHzXEjIkTRIwhaREILrcrbyAvauYllRnbACpdVpd21~XROqPaf-bqcFhxt5H-aiHnuNkmwT3E89pCuSpg7E4h5D~5nu0Gd~HzZ2Cd53jc~3degXMJ-~Ji6ASByNdjKubAtdi5P4o7kiltiPH4jS300S1aifEM6zWWFimw3EGyDlShIbz6y-zMFOY3RyHxR3R9mqmJRvz6EtB8rl~2AyagLiIfWizlQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

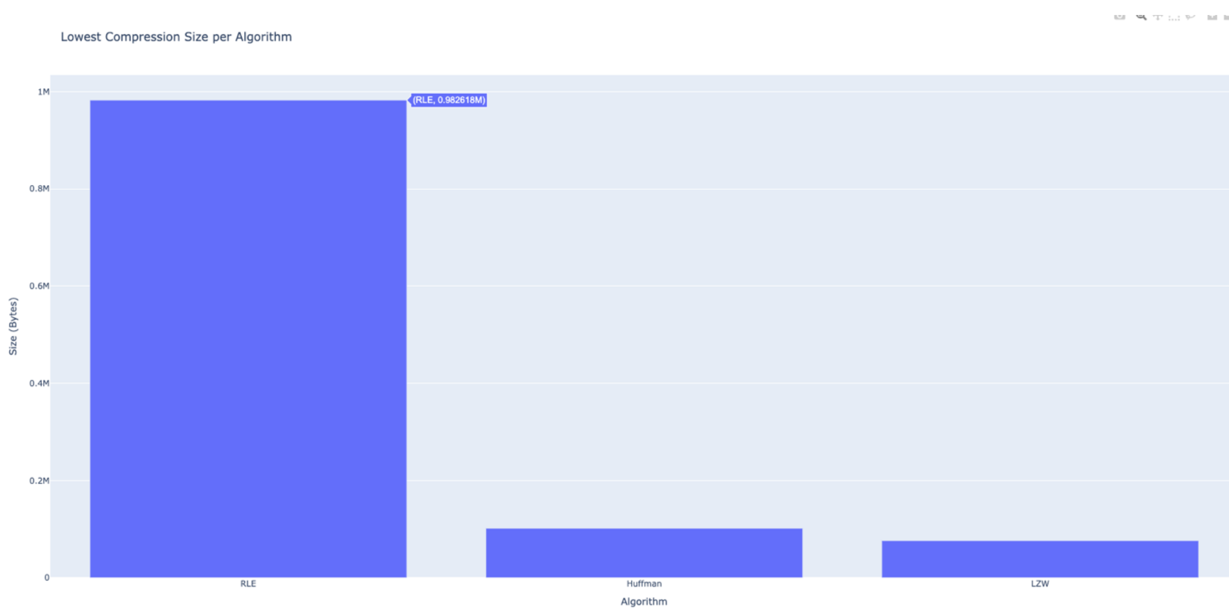
- Huffman Coding performance (used to test version of the algorithm) https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7845058
- Lempel-Ziv-Welch (LZW) https://www.researchgate.net/profile/Asral-Jambek/publication/287236639_Performance_comparison_of_Huffman_and_Lempel-Ziv_welch_data_compression_for_wireless_sensor_node_application/links/5acb16880f7e9bcd5198cea8/Performance-comparison-of-Huffman-and-Lempel-Ziv-welch-data-compression-for-wireless-sensor-node-application.pdf

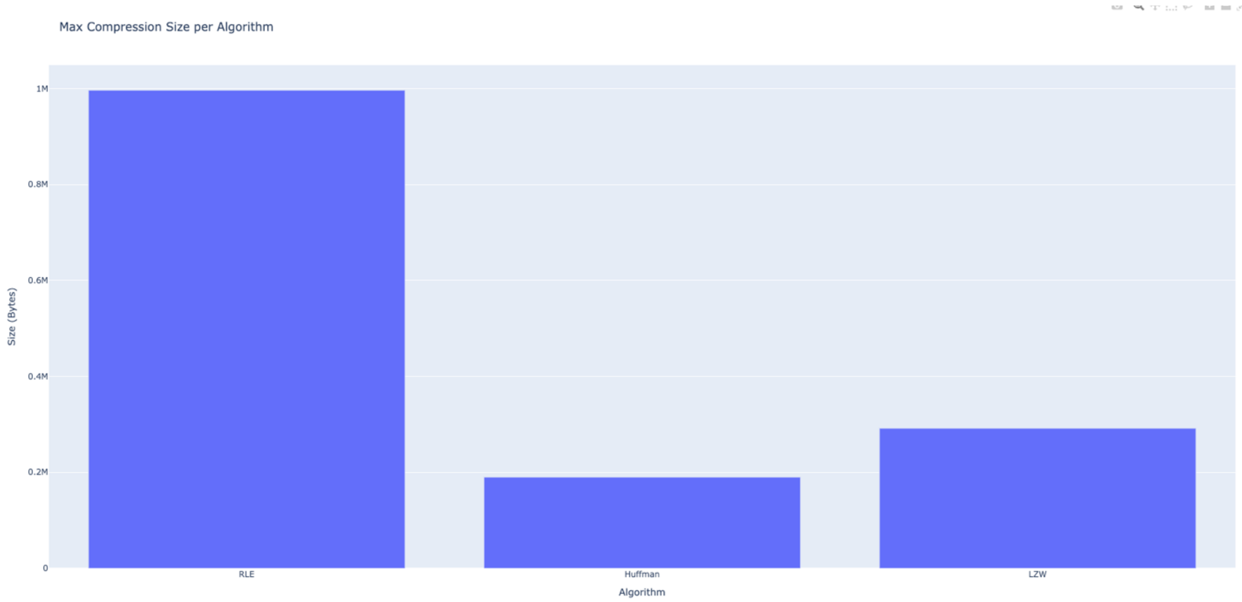
4.

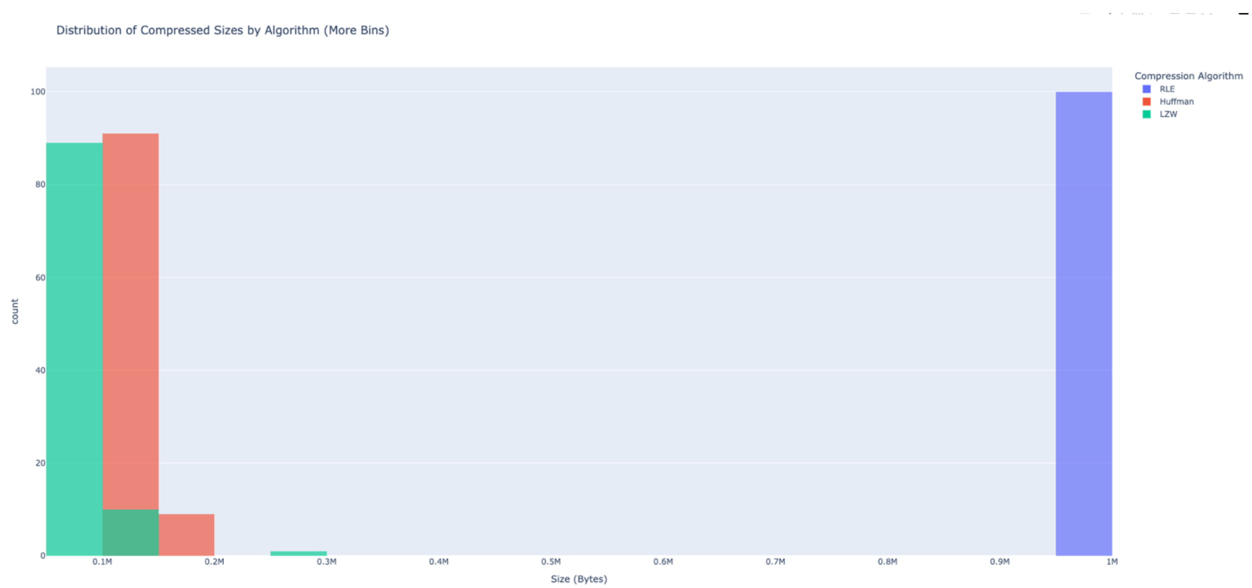

Distribution of Compression Sizes by Algorithm
1M
0.8M
0.6M
0.4M
0.2M
Size (Bytes)
RLE
Huffman
LZW
Compression Algorithm


Average Compression Size per Algorithm
1M
0.8M
0.6M
0.4M
0.2M
0
Size (Bytes)
RLE
Huffman
LZW
Algorithm


Average Compression Time per Algorithm
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Time (Seconds)
RLE
Huffman
LZW
Algorithm


Lowest Compression Size per Algorithm
(RLE, 0.982618M)
1M
0.8M
0.6M
0.4M
0.2M
0
Size (Bytes)
RLE
Huffman
LZW
Algorithm


Max Compression Size per Algorithm
Size (Bytes)
0
0.2M
0.4M
0.6M
0.8M
1M
RLE
Huffman
LZW
Algorithm


Distribution of Compressed Sizes by Algorithm (More Bins)
count
0
20
40
60
80
100
0.1M
0.2M
0.3M
0.4M
0.5M
0.6M
0.7M
0.8M
0.9M
1M
Size (Bytes)
Compression Algorithm
RLE
Huffman
LZW

Compression Results for RLE, Huffman Coding, and LZW

Overview

This analysis evaluates the performance of three compression algorithms—Run Length Encoding (RLE), Huffman Coding, and Lempel-Ziv-Welch (LZW)—on a dataset of audio files. The evaluation includes metrics like average compression size, compression time, and the distribution of compressed sizes (with finer granularity), as well as the lowest and maximum compressed sizes.

Key Findings

1. **Compression Size**
   - **Average Sizes**: Huffman Coding achieved the best compression, followed by LZW. RLE produced the largest compressed sizes, indicating inefficiency for audio data.
   - **Lowest Sizes**: Huffman Coding excelled, achieving the smallest sizes.
   - **Maximum Sizes**: RLE had the largest maximum compressed sizes, while Huffman and LZW maintained tighter bounds.
2. **Compression Time**
   - Huffman Coding took the longest time, reflecting its complexity. LZW offered moderate times, while RLE was the fastest due to its simplicity.
3. **Distribution of Compressed Sizes**

- Huffman and LZW showed consistent and predictable compressed sizes with minimal variance. RLE exhibited a broader range, reaffirming its inefficiency.

Implications

Huffman Coding is ideal for minimizing storage space, while LZW offers a balance between size and speed. RLE, though fast, is unsuitable for non-repetitive data like audio. Analyzing maximum sizes further highlights the reliability of Huffman and LZW for stable compression.

Conclusion

Huffman Coding is the most efficient algorithm for the Neuralink dataset, excelling in compression size and consistency. LZW is a strong alternative for faster processing. Future work could explore hybrid methods combining RLE’s speed with the efficiency of Huffman and LZW.