Collaborative Fog Computing: Fog Head Selection and Data Compression in Internet of EEG-Healthcare Things

BTP Batch : **B25SSR01**

Druva Sai (S20220010222)
Surya Sujit (S20220010217)
Abhinash (S20220010221)

Supervisor
Dr. Sreeja SR
Assistant Professor
CSE Group

Indian Institute of Information Technology Sri City, Tirupathi
Andhra Pradesh

Table of Contents

Introduction

- Internet of EEG Healthcare Things.
  - Electroencephalography (EEG) : Measure electrical activity of the human brain.
  - Advancements in wireless EEGs real-time monitoring
  - Integration of Internet of Healthcare Things (IoHT) improves real-time processing and decision making.
  - IoHT provides efficient data transmission, security , scalability and personalized healthcare insights.
  - Supports Heterogeneity.

Introduction

- Fog computing
  - Supports real-time health care data analytics
  - Reduces data overhead in the healthcare server
  - Provides light weight computation closer to the healthcare devices
  - Helps in parallel and distributed data processing
  - Hence reduces latency , bandwidth

What is FOG computing?

Fog computing is a technology that processes data **closer to where it is created**, instead of sending everything to the cloud. It acts as a **middle layer** between smart devices (like EEG sensors) and cloud servers, helping to analyze and store data faster.

Why is Fog computing Important

1. **Faster Processing** – Data is analyzed locally, reducing delays.
2. **Saves Internet Bandwidth** – Only important data is sent to the cloud.
3. **Works Even with Poor Internet** – Devices can function without full cloud access.
4. **More Secure** – Sensitive data stays closer to the user.

Objectives

Objective 1:

Layered architecture and mathematical modelling for Internet of EEG Healthcare Things Network.

- Introducing fog computing in the IoT network .
- Mathematical modelling of the proposed architecture

Objective 2:

Optimal Fog Head Selection among heterogeneous fog devices

- Multicriteria decision making approach for selection of distinguished fog device as fog head.

Objective 3:

EEG Data Compression for Time-Critical IoT Applications

- Efficient data compression technique using machine learning
- Analysis of compression ratio and compression power

Mathematical Modeling

EEG parameters	Fog parameters	Network parameters
Number of EEG Devices	Number of Fog Devices	EEG and Fog Placement
Number of Channels	Min and Max Clock Frequency	Network Bandwidth
Sampling Rate	Primary and Secondary Memory	Noise in the Medium
Time Slot	Device Energy	Data Propagation Speed
EEG data size	MIPS	
	Number of Cores	

Mathematical Modeling

Delay Model:

$$D_{f_i}^{Total} = D_{f_i}^{trans} + D_{f_i}^{prop} + D_{f_i}^{proc} + D_{f_i}^{queue} \quad (1)$$

In eq(1)

$$D_{f_i}^{trans} = \frac{D_s}{B \times log_2(1 + SNR)} \quad (2)$$

$$D_{f_i}^{prop} = \frac{dist_{EEG,K} + dist_{K,f_i}}{V} \quad (3)$$

$$D_{f_i}^{proc} = \frac{n}{C_{f_i}^{avg}} \quad (4)$$

$$D_{f_i}^{queue}(T^n) = D_{f_i}^{proc}(T^1) + D_{f_i}^{proc}(T^2) + \cdots + D_{f_i}^{proc}(T^{n-1}) \quad (5)$$

D_s : data size
B : bandwidth of the network (Hz)
SNR : Signal-to-Noise Ratio
$dist_{EEG,K}$: distance between EEG and Fog head
$dist_{K,f_i}$: distance between Fog head and i^{th} Fog device
V : propagation speed of the signal in the medium
n : cpu cycles required to process 1bit of data
$C_{f_i}^{avg}$: Average clock frequency
T^n : n^{th} task in the queue

Mathematical Modeling

Energy Model:

$$E_{f_i}^{Total} = E_{f_i}^{proc} + E_{f_i}^{trans} + E_{f_i}^{idle} \quad (6)$$

$$E_{f_i}^{proc} = e^{proc} \times D_{f_i}^{proc} \quad (7)$$

$$E_{f_i}^{trans} = e^{trans} \times D_{f_i}^{trans} \quad (8)$$

e^{proc} : energy consumption for processing

e^{trans} : energy consumption for transmitting

Mathematical Modeling

Performance and storage Model :

$PM_{f_i}^{occ}$: Primary memory occupied

$SM_{f_i}^{occ}$: Secondary memory occupied

$C_{f_i}^{avg}$: Average clock frequency

$$SE = \alpha \times \left(\frac{PM_{f_i}^{occ}}{\mathrm{PM}}\right) + \beta \times \left(\frac{SM_{f_i}^{occ}}{\mathrm{SM}}\right) \quad (9)$$

$$CPI = \frac{C_{fi}^{avg}}{MIPS \times 10^6} \quad (10)$$

Dataset Description

Bonn EEG Dataset

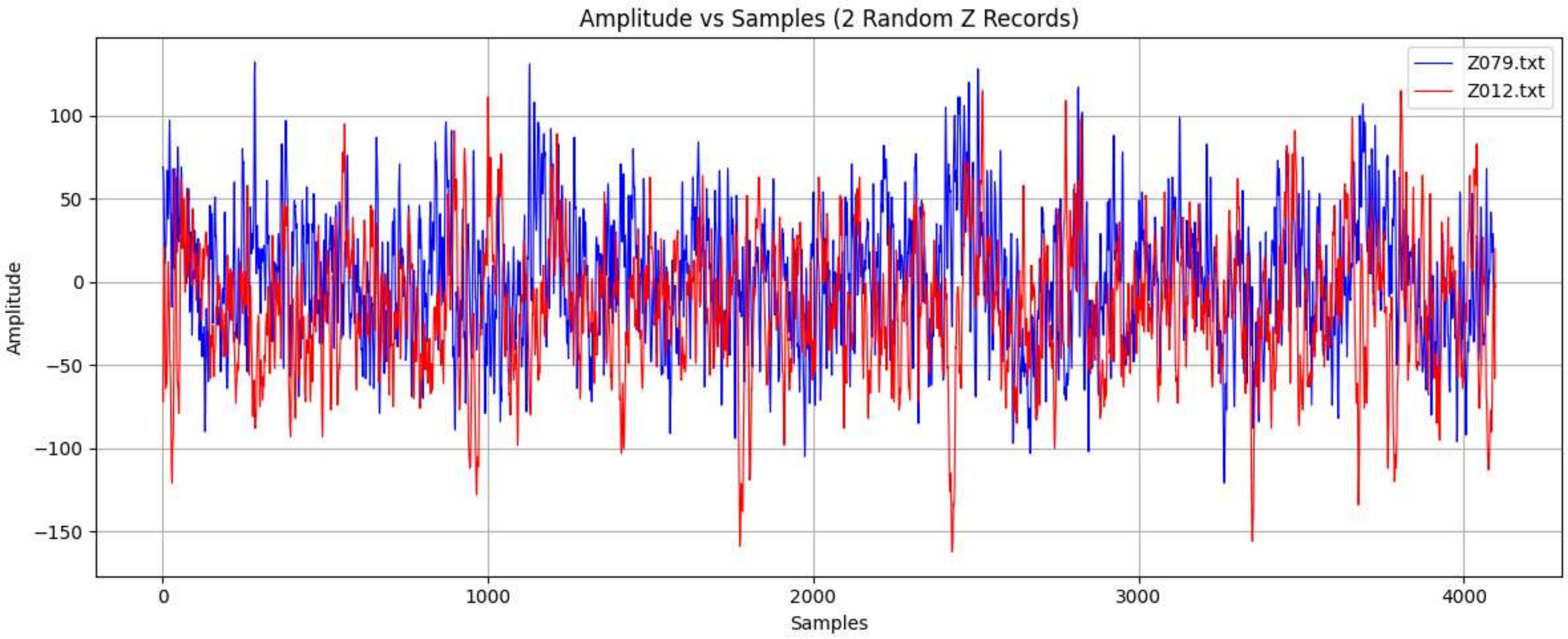
Content:
EEG recordings from healthy people and epilepsy patients.

Structure:

- **5 Subsets** :
  - **F, N**: Scalp EEG from 5 healthy people (eyes open/closed).
  - **O, S**: Brain EEG from 5 epilepsy patients (no seizure, from seizure/non-seizure areas).
  - **Z**: Brain EEG during seizures.
- **Each Subset**: 100 single-channel EEG segments, each 23.6 seconds long.
- **Channels**: One channel per segment, chosen from multi-channel EEG (10–20 system for scalp).
- **Sampling Rate**: 173.61 Hz (records 173.61 data points per second).
- **Bandwidth**: 0.5–85 Hz.
- **Total Segments**: 500 (100 per subset).
- **Format**: Text files (100 TXT files per subset, stored in ZIP archives).

Comparing Two EEG Signals

This graph shows two brain wave signals:



Metrics for EEG Compression

Metric Name	Meaning	Formula
Compression Percentage (CP %)	How much we shrunk the EEG file	CP = (1 - 1/CR) × 100
Compression Ratio (CR)	How many times smaller the file is	CR = Original Size (bytes) / Compressed Size (bytes)
Compression Time (s)	How long it takes to shrink the file	Compression Time = End Time - Start Time (seconds)
Decompression Time (s)	How long it takes to unshrink the file	Decompression Time = End Time - Start Time (seconds)
Compressed Size (bits)	How big the shrunk file is	Compressed Size = Length of Compressed Bits

EEG Compression

K-Means Clustering Only

- Reduces the EEG signal to a small number of representative values (centroids) by grouping similar samples.
- Lowers the number of unique values, providing some data reduction.
- Maintains the original sequence of the signal, so redundancy in local temporal structure is not addressed.
- Compression achieved is limited because similar values may still be spread out in the data.
- Essential features are preserved, but efficiency is less compared to a combined approach.

Delta Encoding Only

- Replaces each sample with the difference from its previous value, highlighting changes in the signal.
- Works well for smooth regions, creating many small values.
- For highly variable or noisy EEG signals, large delta values remain, which limits compressibility.
- Does not reduce the diversity of unique signal values up front.
- Achieves only modest compression—especially for complex biomedical signals on its own.

EEG Compression

Delta And Arithmetic Coding

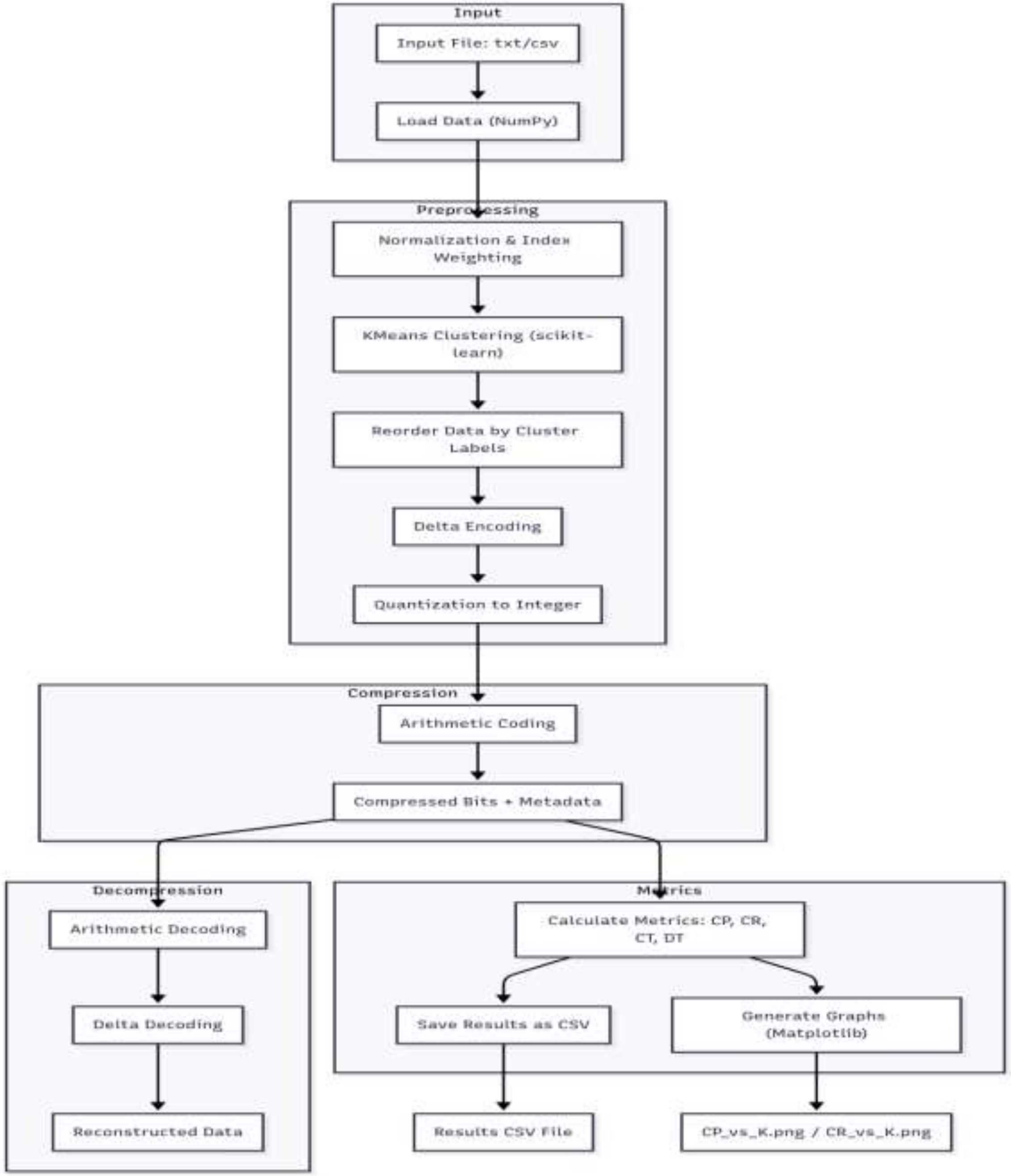
- Delta encoding processes the EEG signal by storing only the difference between consecutive samples, producing a sequence of smaller, more compressible values.
- The delta values are quantized to a fixed number of bits, standardizing their representation.
- Arithmetic coding then uses the probability distribution of these quantized delta values to encode them into a highly compact bitstream.
- This combination achieves stronger compression than delta encoding alone, because arithmetic coding efficiently removes statistical redundancy by assigning shorter codes to common delta values.
- During decompression, arithmetic decoding reconstructs the quantized deltas, and delta decoding uses cumulative summation to perfectly restore the original EEG waveform.
- The overall process preserves the essential structure of the signal while yielding high compression efficiency.

Proposed Model

K-Means + Delta + Arithmetic

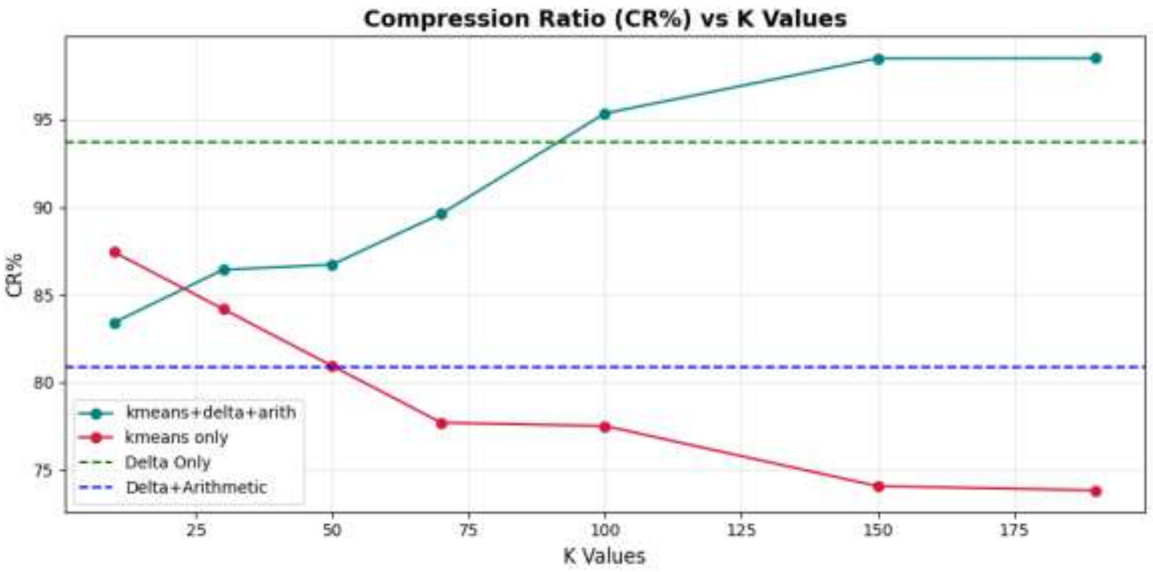
- K-Means + Delta + Arithmetic compression combines three techniques to achieve very high compression on EEG data.
- First, K-Means clustering groups similar signal values into K clusters, replacing each data point with its cluster centroid.
- This reduces the number of unique values and helps reorganize data so similar samples sit next to each other.
- Then, delta encoding is applied to the reordered data, storing only the small differences between consecutive samples, which are easier to represent compactly.
- Finally, the quantized delta values are compressed with arithmetic coding, which assigns shorter codes to more frequent differences for optimal entropy-based compression.
- Together, this pipeline minimizes redundancy, drastically reduces storage size, and preserves the essential EEG waveform for later reconstruction.

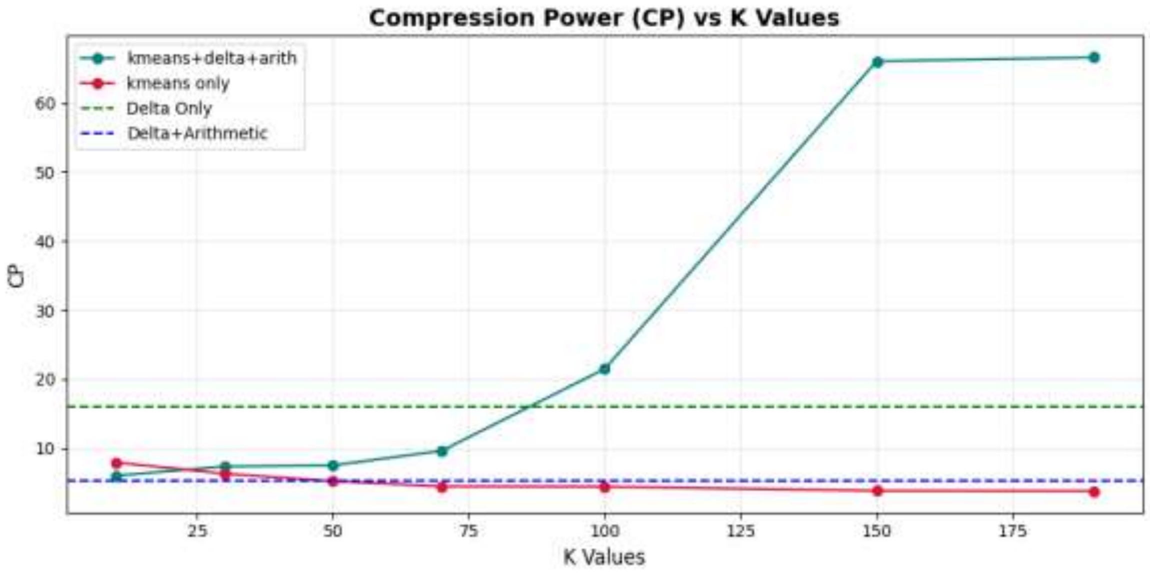
Architecture


Input
Input File: txt/csv
Load Data (NumPy)
Preprocessing
Normalization & Index Weighting
KMeans Clustering (scikit-learn)
Reorder Data by Cluster Labels
Delta Encoding
Quantization to Integer
Compression
Arithmetic Coding
Compressed Bits + Metadata
Decompression
Arithmetic Decoding
Delta Decoding
Reconstructed Data
Metrics
Calculate Metrics: CP, CR, CT, DT
Save Results as CSV
Generate Graphs (Matplotlib)
Results CSV File
CP_vs_K.png / CR_vs_K.png

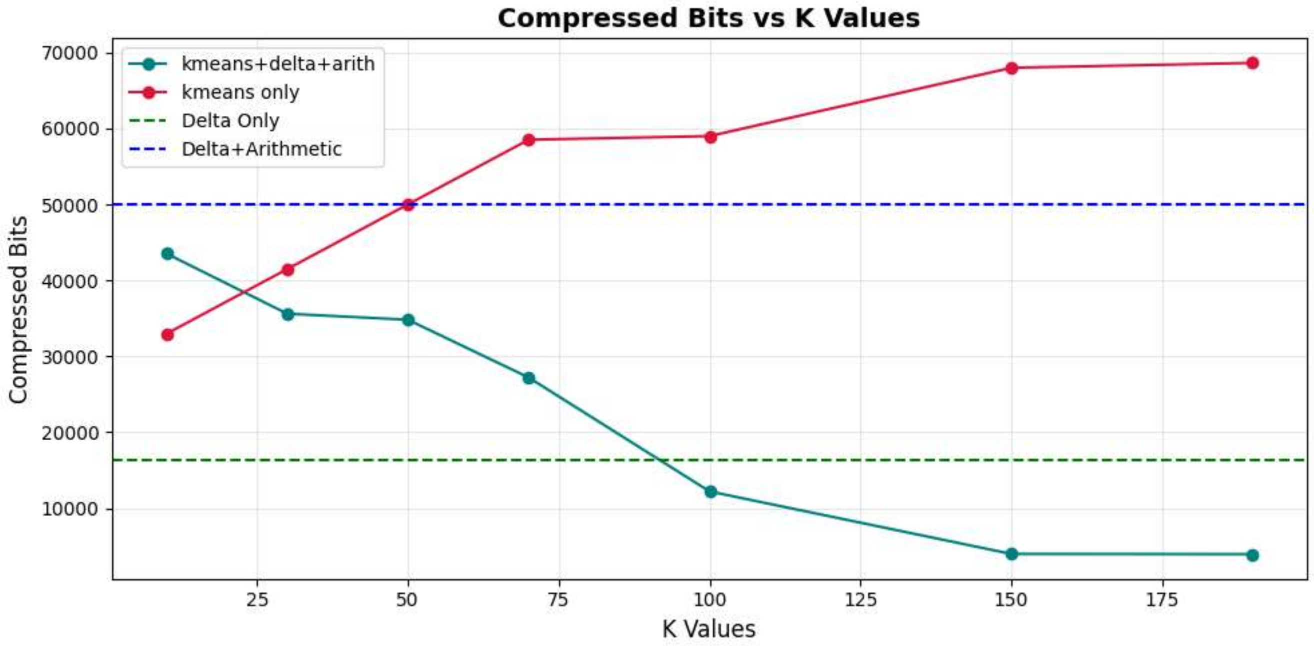
Results

Performance Metrics

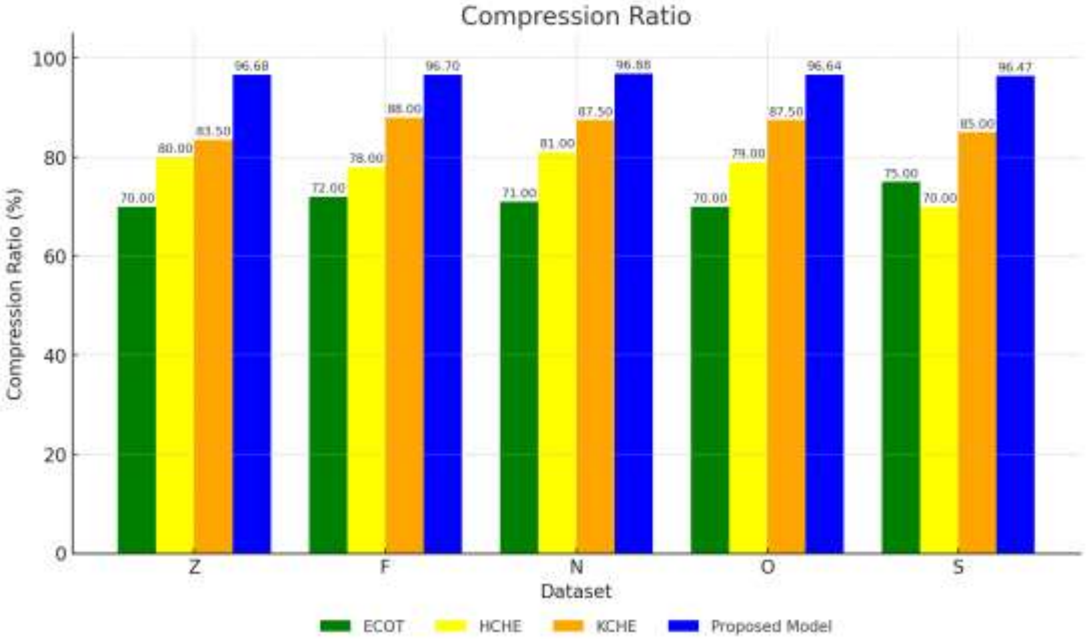


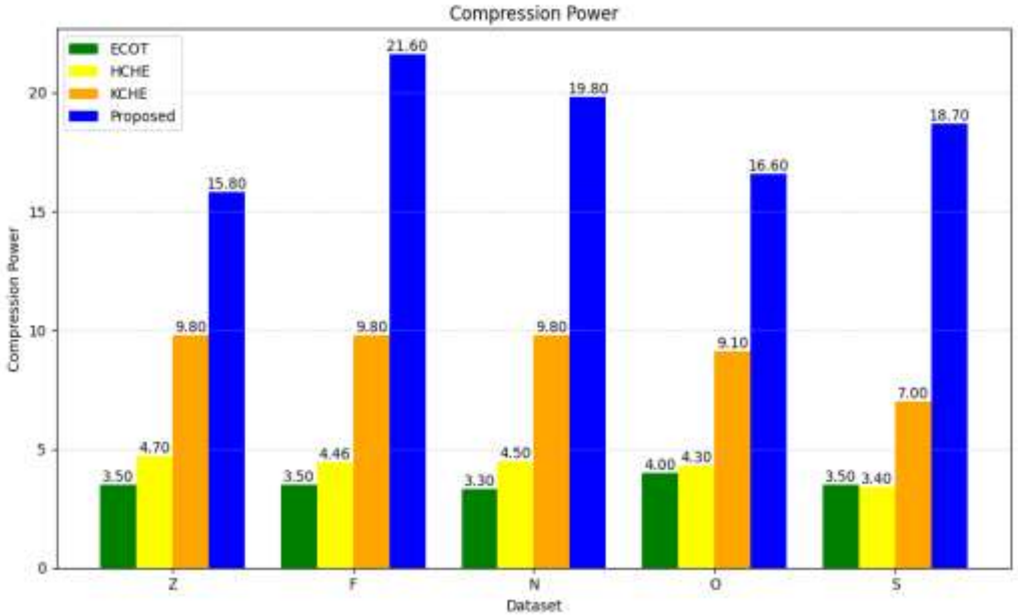


Performance Metrics

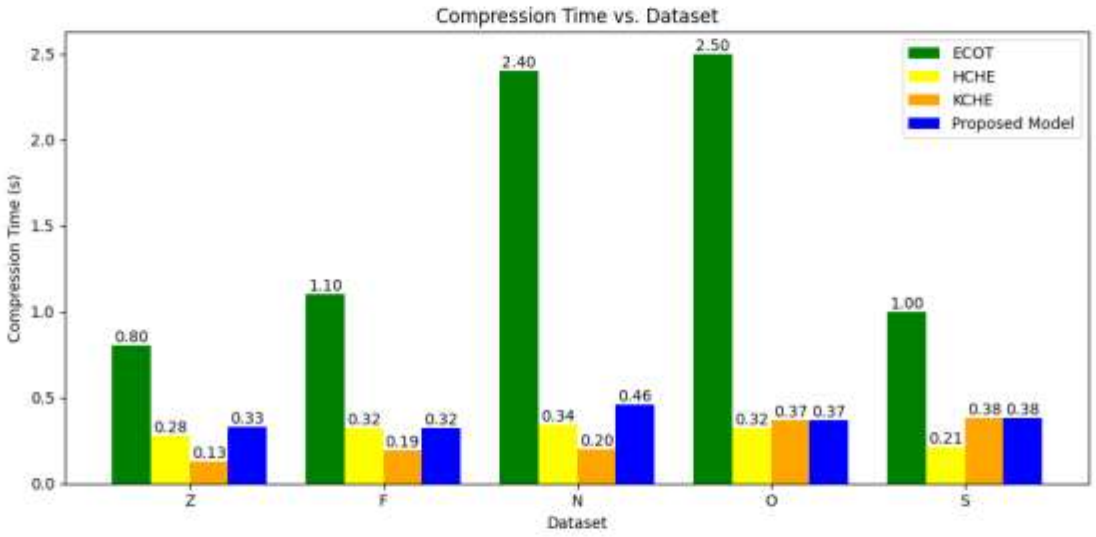


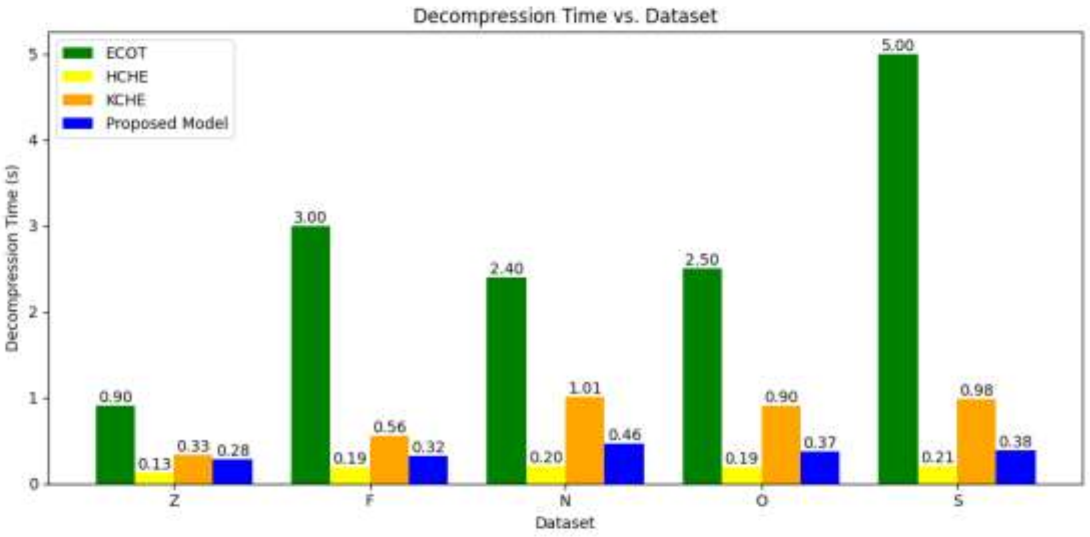
Comparative Analysis of Compression Metrices Across Models





Comparative Analysis of Compression Metrices Across Models





Current Work

Multi-Criteria Decision-Making (MCDM) approach will be implemented to determine the Optimal selection of fog heads, considering multiple performance criteria to enhance Overall System efficiency and effectiveness.

Multi-Criteria Decision-Making (MCDM)

What is Multi-Criteria Decision Making (MCDM)?

- Provides a systematic way to compare several alternatives when multiple factors must be considered.
- Allows decision-makers to evaluate conflicting criteria (such as speed versus cost).
- Offers a clear structure for making informed and balanced choices.

Why use MCDM in Complex Systems?

- In fog computing, key factors like latency, energy consumption, and bandwidth utilization all impact system performance.
- MCDM helps balance the trade-off between fast response (low latency), low energy use, and efficient bandwidth, which often conflict with each other.
- Using MCDM ensures that the final decision reflects all important criteria from the system's data, not just a single metric or intuition.

DEMATEL Method

What is DEMATEL?

- DEMATEL stands for Decision Making Trial and Evaluation Laboratory—a method for analyzing complex relationships among decision criteria.
- Helps identify which criteria are the most influential and which are most affected in a system.
- Uses diagrams and matrices to visualize cause-effect links between factors.

Why use DEMATEL in Fog Computing?

- In fog computing, factors like latency, energy consumption, and bandwidth shown in your data are often interconnected and can influence one another.
- DEMATEL clarifies which criteria (e.g., minimizing latency or saving energy) have the biggest impact on overall system performance.
- By separating influential (cause) and dependent (effect) criteria, DEMATEL guides resource allocation and helps prioritize improvements that matter most for the system.

MOORA Method

What is MOORA?

- MOORA (Multi-Objective Optimization by Ratio Analysis) is an approach for ranking alternatives using multiple criteria.
- Converts different metrics to a dimensionless scale, allowing fair comparisons between fog heads on factors like latency, energy, and bandwidth.
- Provides a simple way to aggregate scores and rank fog head options objectively.

MOORA's Strengths for Fog Head Selection

- Handles diverse criteria from your data, regardless of units or scales.
- Produces an overall ranking, highlighting the best-performing fog heads for your system needs.

Dataset

Fog device	Fog Processor	C min	C max	No of cores	MIPS	Primary Memory	Secondary Memory (GB)	Total energy (Joules)	Unit Energy consumption	Idle energy
1	F1(Nvidea Jetson Nano)	1.43	1.43	4	9000	4	64	335700	93.25	16785
2	F2(Raspberry Pi 4 Model B)	1.5	1.5	4	12000	4	128	90000	25	4500
3	F3(NVIDIA Jetson Xavier NX)	1.4	1.9	6	9000	8	128	335700	93.25	16785
4	F4(Intel Celeron N4000)	1.1	2.6	2	8800	4	32	162000	45	8100
5	F5(Snapdragon 888)	1.8	2.84	8	12000	8	32	234000	65	11700
6	F6(A15 Bionic chip)	1.8	3.23	6	15000	6	128	72000	20	3600
7	F7(AMD Ryzen 3 2200U)	2.5	3.4	2	20000	4	128	126000	35	6300
8	F8(AMD Ryzen 3 5300U)	2.6	3.8	4	12080	8	128	162000	45	8100

DEMATEL in Fog Head Selection

Role of DEMATEL in Selecting Fog Heads

- DEMATEL helps reveal how criteria from your dataset—such as fog head processing speed, energy use, and network load—affect each other.
- By analyzing these relationships, DEMATEL highlights which fog heads align best with the most influential criteria.
- Makes it easier to spot key trade-offs and dependencies in the actual selection process.

Benefits for System Efficiency

- Focuses attention (and resources) on improving the criteria that drive system performance.
- Adds transparency: decision-makers can see which factors should guide fog head selection, based on real system data.

DEMATEL in Fog Head Selection

Step 1: Identification of Criteria

Based on the fog device dataset, the following nine criteria are considered:

C_1: Minimum Clock Speed (Cmin_GHz)

C_2: Maximum Clock Speed (Cmax_GHz)

C_3: Number of Cores

C_4: MIPS

C_5: Primary Memory (GB)

C_6: Secondary Memory (GB)

C_7: Total Energy Consumption (Joules)

C_8: Unit Energy Consumption

C_9: Idle Energy

These criteria jointly influence fog head performance, energy efficiency, and overall system effectiveness.

DEMATEL in Fog Head Selection

Step 2: Construction of the Direct-Relation Matrix

A **direct-relation matrix (X)** of size 9 × 9 is constructed to represent the degree of influence among criteria.

Each element xijx_{ij}xij denotes the influence of criterion iii on criterion jjj.

A discrete scale is used:

- 0 – No influence
- 1 – Low influence
- 2 – Medium influence
- 3 – High influence
- 4 – Very high influence

All diagonal elements are set to zero since a criterion does not influence itself.

This matrix captures expert knowledge and system understanding, such as:

Higher clock speeds and core counts increasing MIPS.

Increased computational power leading to higher energy consumption.

Energy-related criteria being mostly dependent on performance-related criteria.

DEMATEL in Fog Head Selection

Step 3: Normalization of the Direct-Relation Matrix

To ensure numerical stability and comparability, the direct-relation matrix is normalized.

Let

$$\alpha = \frac{1}{\max\left(\sum_{j=1}^{n} x_{ij}\right)}$$

The normalized matrix NNN is obtained as:

$$N = \alpha \times X$$

This step ensures that all elements of NNN lie in the range [0,1].

DEMATEL in Fog Head Selection

Step 4: Calculation of the Total-Relation Matrix

The total-relation matrix T accounts for both **direct and indirect influences** among criteria and is computed as:

$$T = N(I - N)^{-1}$$

where:

- I is the identity matrix.
- $(I - N)^{-1}$ incorporates indirect influence paths.

Matrix T represents the overall influence structure of the system.

DEMATEL in Fog Head Selection

Step 5: Computation of D and R Values

From matrix T:

- **D (Driving Power):**

$$D_i = \sum_{j=1}^{n} t_{ij}$$

Indicates how much criterion i influences other criteria.

- **R (Dependence Power):**

$$R_i = \sum_{j=1}^{n} t_{ji}$$

Indicates how much criterion i is influenced by others.

DEMATEL in Fog Head Selection

Step 6: Prominence and Relation Analysis

Using D and R:

- **Prominence:** D_i+R_i

Represents the overall importance of a criterion in the system.

- **Relation:** D_i-R_i
- Indicates the role of the criterion:
- Positive value → **Cause (driving criterion)**
- Negative value → **Effect (dependent criterion)**

This classification helps separate performance-driving factors from outcome-dependent factors.

Step 7: Calculation of DEMATEL Weights

The prominence values $(D_i + R_i)$ are normalized to obtain the final weights:

$$w_i = \frac{(D_i + R_i) - \min(D+R)}{\sum_{k=1}^{n}[(D_k + R_k) - \min(D+R)]}$$

These weights reflect the relative importance of each criterion and are later used in MOORA for ranking fog heads.

DEMATEL in Fog Head Selection

Step 8: Interpretation for Fog Head Selection

- Criteria such as **maximum clock speed, MIPS, and energy consumption** exhibit higher prominence, indicating strong influence on system performance.
- Energy-related factors are mainly effect-type criteria, dependent on computational parameters.
- The derived weights enable objective, influence-aware decision-making rather than assumption-based selection.

DEMATEL Final Weights

Criterion	D_influence	R_received	Prominence(D+R)	Relation(D-R)	Weight_from_DEMATEL
Cmin_GHz	1.016419	0.160594	1.177013	0.855825	0.117219
Cmax_GHz	1.380942	0.766532	2.147473	0.61441	0.240165
Cores	0.865872	0	0.865872	0.865872	0.077801
MIPS	0.552931	1.041835	1.594767	-0.488904	0.170143
PrimaryMem_GB	0.362348	0	0.362348	0.362348	0.01401
SecondaryMem_GB	0.251762	0	0.251762	0.251762	0
TotalEnergy_J	0.216449	1.427678	1.644127	-1.211228	0.176396
UnitEnergy	0.216449	0.979757	1.196206	-0.763308	0.11965
IdleEnergy	0.216449	0.703227	0.919676	-0.486777	0.084617

MOORA in Fog Head Selection

How MOORA Works with Fog Head Data

- Each fog head is scored across all relevant criteria from your dataset (e.g., speed, energy, network usage).
- MOORA uses weights (from DEMATEL analysis) to reflect the importance of each criterion in the final ranking.
- Ensures that fog heads meeting the most critical system requirements come out on top.

Advantages Over Simple Selection

- Integrates strengths and weaknesses from actual system data, rather than focusing on a single metric.
- Results in selection that is both evidence-based and tailored to real performance needs.

Integrated DEMATEL-MOORA Approach

Combining DEMATEL and MOORA for Optimal Selection

- DEMATEL first identifies and prioritizes criteria from your fog head data based on their impact and interdependencies.
- MOORA then evaluates all fog head options using those prioritized criteria and their weights.
- The approach ensures the selected fog heads offer the best balance across all important metrics, according to system data.

Value for Decision Makers

- Promotes transparent, data-driven, and well-justified choices.
- Maximizes system effectiveness by systematically accounting for every major performance factor reflected in your fog computing environment.

MOORA: Step-by-Step Procedure

1.Build decision matrix

Rows = processors (F1…F8)

Columns = criteria (clock speed, energy, etc.)

2.Vector normalization

Each value is divided by the square-root of sum of squares for that column

Converts all metrics to a dimensionless scale (fair comparison)

3.Apply weights

Multiply each normalized column by its DEMATEL weight

Gives a weighted normalized matrix

4.Compute MOORA score for each processor

Sum of weighted benefit criteria – Sum of weighted cost criteria

5.Rank processors

Higher MOORA score ⇒ better overall alternative

Benefit vs Cost Criteria

Benefit (higher is better):

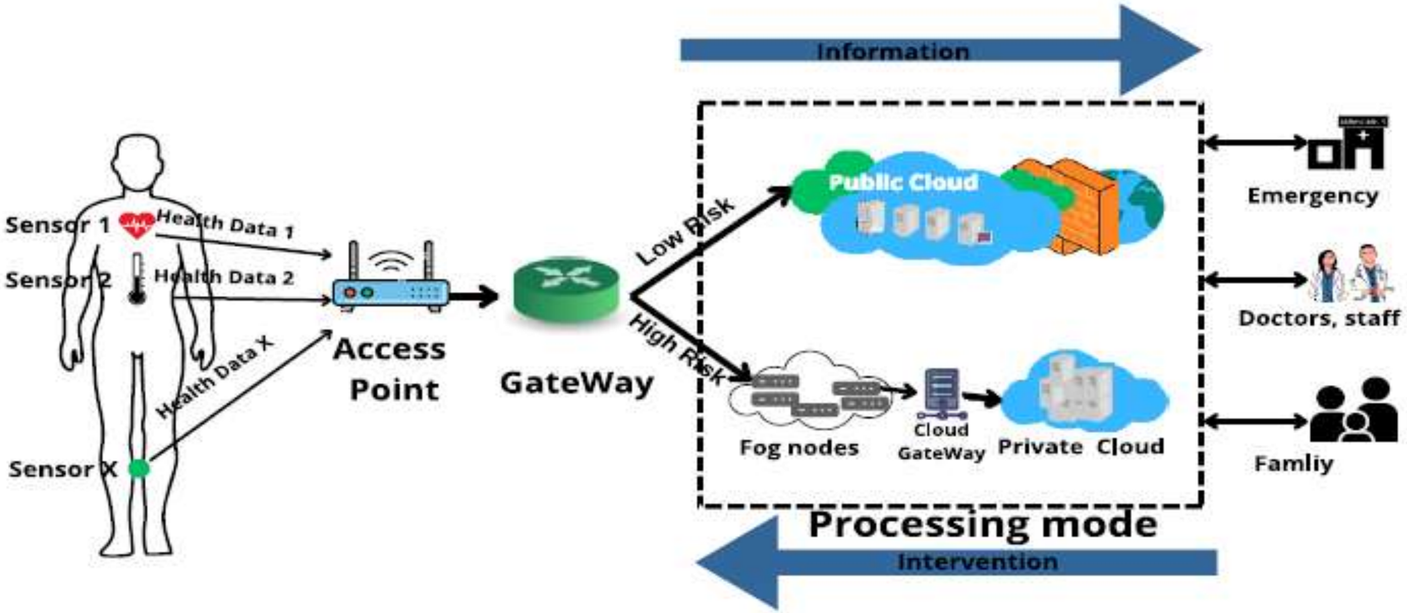
- Cmin_GHz (minimum clock speed)
- Cmax_GHz (maximum clock speed)
- Cores
- MIPS
- PrimaryMem_GB
- SecondaryMem_GB

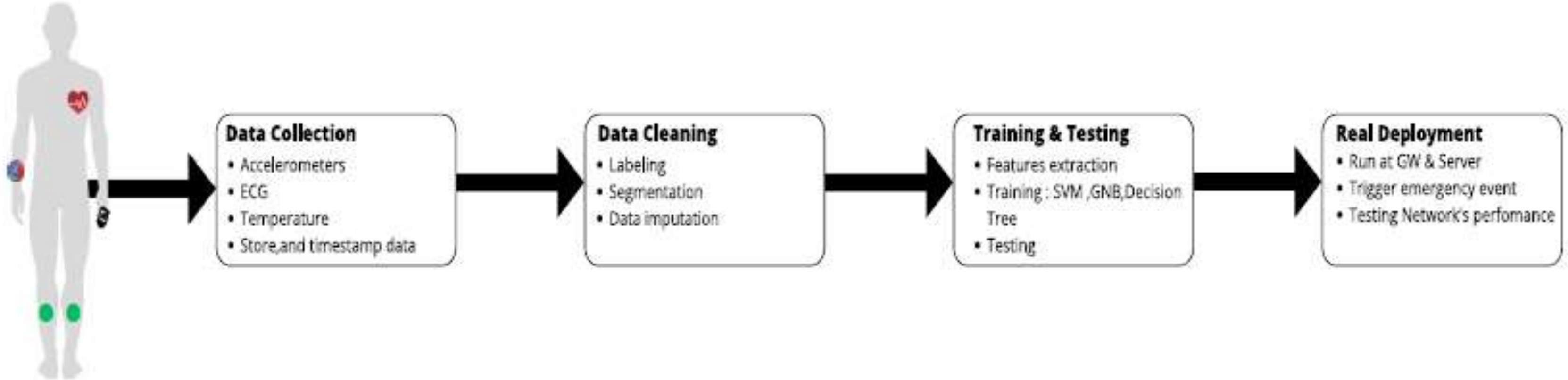
Cost (lower is better):

- TotalEnergy_J
- UnitEnergy
- IdleEnergy
- MOORA treats **benefits as positive** and **costs as negative** in the final score.

Final MOORA Results

```
=== Final MOORA Ranking ===
               Processor  MOORA_score  MOORA_rank  Relative_Performance_%
F6       A15 Bionic chip     0.205362           1              100.000000
F7      AMD Ryzen 3 2200U    0.191810           2               93.401015
F8      AMD Ryzen 3 5300U    0.160886           3               78.342675
F2   Raspberry Pi 4 Model B  0.106145           4               51.686959
F5         Snapdragon 888    0.089365           5               43.515636
F4     Intel Celeron N4000   0.059555           6               28.999743
F3  NVIDIA Jetson Xavier NX -0.038774           7              -18.880868
F1      Nvidia Jetson Nano  -0.067281           8              -32.762132
```


Information
Sensor 1
Health Data 1
Sensor 2
Health Data 2
Health Data X
Sensor X
Access Point
GateWay
Low Risk
High Risk
Public Cloud
Fog nodes
Cloud GateWay
Private Cloud
Emergency
Doctors, staff
Famliy
Processing mode
Intervention


Data Collection
• Accelerometers
• ECG
• Temperature
• Store,and timestamp data
Data Cleaning
• Labeling
• Segmentation
• Data imputation
Training & Testing
• Features extraction
• Training : SVM ,GNB,Decision Tree
• Testing
Real Deployment
• Run at GW & Server
• Trigger emergency event
• Testing Network's perfomance

References

- Idrees, A. K., Idrees, S. K., Couturier, R., & Ali-Yahiya, T. (2022). An edge-fog computing-enabled lossless EEG data compression with epileptic seizure detection in IoMT networks. *IEEE Internet of Things Journal*.

- Khlief, M. S., & Idrees, A. K. (2022). Efficient EEG data compression technique for Internet of Health Things networks. *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*.

- Idrees, S. K., & Idrees, A. K. (2022). New fog computing enabled lossless EEG data compression scheme in IoT networks. *Journal of Ambient Intelligence and Humanized Computing, 13*, 3257–3270.

- Idrees, A. K., & Khlief, M. S. (2023). Efficient compression technique for reducing transmitted EEG data without loss in IoMT networks based on fog computing. *The Journal of Supercomputing, 79*, 9047–9072.

References

- Idrees, A. K., & Khlief, M. S. (2023). Lossless EEG data compression using clustering and encoding for fog computing based IoMT networks. *International Journal of Computer Applications in Technology*, 72(1), 77–83.
- Karimu, R. Y., & Azadi, S. (2016). Lossless EEG compression using the DCT and Huffman coding. *Journal of Scientific & Industrial Research, 75*(10), 615-620.

Thank You