Journal of Digital Information Management

DLINE JOURNALS

ISSN: 0972-7272

JDIM 2026: 24 (3)

Benchmarking the Decoherence Threshold of Hybrid Quantum Classical Networks for Edge based EEG Telemetry

Sasiram Anupoju
ACE Engineering College
Ankushapur, Telangana 500031. India
sasiram.anupoju@aceec.ac.in

Kavitha Soppari
ACE Engineering College
Ankushapur, Telangana 500031. India
kavithasoppari@aceec.ac.in

ABSTRACT

This paper studies how noise and decoherence affect a hybrid quantum classical neural network (HQCNN) when it is applied to EEG based eye state classification, a task that closely mirrors what an edge based brain-computer interface would need to do in the field. Quantum machine learning holds real promise for embedded health systems, but the hardware of today is noisy, and nobody has clearly mapped out exactly where the accuracy starts to collapse. We built a model that uses a four qubit parameterized quantum circuit implemented in PennyLane and backed by a PyTorch classical front end, then trained it on the wellknown EEG Eye State dataset from OpenML. Once training was done, we deliberately injected depolarizing channel noise at six different error rates from a clean 0.00 all the way up to 0.15 and measured what happened to classification accuracy at each step. The results gave us a clear picture. The noiseless model reached 90.75% accuracy. That number held reasonably steady through low noise, but at a depolarizing error rate of around 0.12 the accuracy dropped below the 90% threshold of the baseline, landing at 86.00%, and by 0.15 it had fallen further to 83.75%. We define a hard mathematical threshold as a 10% relative degradation from the noiseless baseline (0.8168). However, the model did not reach this absolute floor at the maximum tested noise of 0.15 (83.75%). Instead, we identify 0.12 as the practical decoherence warning threshold, defined by a sharp inflection in the rate of change (slope) of the accuracy drop. This distinction clarifies that while the model remains mathematically above the 10% degradation floor, 0.12 represents the critical point where hardware engineers must intervene before clinical deployment. The work proves that a compact four qubit circuit can remain useful for EEG classification even under moderate hardware noise, while also making it clear that current NISQ devices would need error mitigation strategies before being trusted in a clinical edge deployment. The findings contribute a concrete benchmark that system designers can use when deciding whether quantum co-processors belong in the next generation of wearable EEG telemetry hardware.

Subject Categories and Descriptors: [**I.5 Pattern Recognition**]; Neural nets: **[C.2.1 Network Architecture and Design**]; Network communications

General Terms: Quantum Machine Learning, Hybrid Quantum Classical Neural Network (HQCNN), Decoherence / Noise Threshold, EEG / Electroencephalography Telemetry Brain Computer Interface (BCI), NISQ

(Noisy Intermediate Scale Quantum) Devices, Edge AI / Edge Computing, Parameterized Quantum Circuit (PQC), Depolarizing Channel Noise, Error Mitigation

Keywords: Hybrid Quantum Classical Neural Network, Decoherence Threshold, EEG Telemetry, De-polarizing Channel Noise, Penny Lane, NISQ Devices, Brain-Computer Interface, Edge AI, Quantum Machine Learning, Parameterized Quantum Circuit

Received: 21 April 2026, Revised: 27 June 2026, Accepted: 6 July 2026

Review Metrics: Review Scale: 0-6, Review Score: 4.7, Inter-Reviewer Consistency: 88.4%

DOI: https://doi.org/10.6025/jdim/2026/24/3/158-165

1. Introduction

Hospitals have recorded brain electrical activity with EEG for a long time. What is newer is the idea of shrinking that whole process sensor, processor, classifier down to a wearable device that fits on someone's head and makes decisions on its own without sending anything to a server. Researchers call this edge based EEG telemetry, and it has some genuinely attractive properties. Low latency is achieved by eliminating the round trip to the cloud. The privacy implications are better for the same reason. And we can use it anywhere, not just where we have a reliable wireless signal. Classical machine learning can handle the classification part well enough when the hardware has the spare compute cycles. The difficulty is that wearable devices run on tiny batteries and have to stay physically small, so there is a hard ceiling on how much processing we can pack in. That constraint is what makes quantum co-processors interesting in this context. Even a very modest quantum circuit, a handful of qubits, might handle certain signal classification problems more economically than a classical network of similar functional capacity. No one has fully proven this at scale, yet the hardware is still maturing, but it is credible enough to be worth investigating seriously. The snag, as always with quantum hardware, is noise. Aqubit is not like a transist or that either holds avolt age or does not. It can be in a super position state that gradually collapses due to its thermal environment, nearby electromagnetic interference, or imperfect gate operations. Physicists group all of this under the umbrella term decoherence. For a learning model, decoherence means that the quantum part of the computation slowly stops doing what the training intended it to do. The model's predictions drift. And the field's current error mitigation approaches, while clever, add computational overhead that a small wearable device probably cannot afford. So what we actually need to know is: where does the classification break down? At what error rate does a hybrid quantum classical EEG classifier go from useful to unreliable? That specific question has not been answered empirically for this type of task. We address it by training a model in clean conditions and then sweeping through a range of injected noise levels to map out the accuracy curve. We call the point where accuracy meaningfully degrades the architecture's decoherence threshold. Our goal is not to claim that quantum methods are already better than classical ones here, they are not, at this scale.

2. Literature Review

2.1 Quantum Machine Learning

Biamonte and his team [3] published a survey about quantum machine learning in 2017. At that time, quantum machine learning was still mostly an idea that people were talking about. They were clear about the limitations of quantum machine learning. Just because quantum computers can do some things faster in a sense, it does not mean they can always handle real world problems. This warning has turned out to be true. People started working on more practical approaches that combined quantum and classical computers. Schuld and Killoran [13] in 2019 and Havlicek [6] and his colleagues at IBM in 2019 suggested ways to use quantum computers for certain problems. They proposed using quantum computers to help with classification problems that could be solved with the computers that would be available soon. What they established was an idea for designing these systems: use classical computers to do the hard work and use quantum computers only for the parts where they can really help. The review of circuits by Cerezo and his team [4] in 2021 is probably the most detailed study on whether these circuits can be trained. Their analysis of the problems that happen when the gradients get too small and the training stops, helped us decide to keep our circuit simple. [5, 7, 9]. If we make the circuit too complicated with many quantum bits, the gradients become extremely small and we cannot learn anything useful from the quantum machine learning. Abbas and his team [1] in 2021 added another argument that quantum models can represent information with fewer parameters than classical models of similar size. Whether this actually works with data is still an open question, but it gives us a good reason to try using even small quantum circuits for quantum machine learning.

2.2 EEG Classification and Brain-Computer Interfaces

Roy [12] and his colleagues did a thorough review of the deep learning literature on EEG analysis in 2019. The features the computer learned on its own outformed those people created. Convolutional networks worked well for a lot of tasks like detecting what people are imagining and figuring out what stage of sleep they are in. The field of EEG analysis has come a long way from the old ways of processing signals. The EEG Eye State dataset, which was put together by Rosler and Suendermann [11] in 2013, became a test case. This is because it is easy to use and still similar to real recordings. Depending on the classifier and the steps taken to prepare the data, the accuracy of the results canbe anywhere from around 85% to over 97%. On the side of using EEG in devices, Zhang [17] and his colleagues worked on making EEG classifiers small enough to fit on wearable hardware in 2020. They found a way to make the network smaller by cutting out parts that are not needed and using data to represent the numbers. This worked well because the model got much smaller and it did not lose much accuracy. The amount of energy that a typical wearable EEG device uses is very low. [10, 14, 15]. This is a problem that keeps researchers looking for new ways to do things including, using special co-processors that use something called quantum technology, even when the oldways are still a little bit better.

3. Methodology

3.1 Dataset and Preprocessing

We used the EEG Eye State dataset obtained from the OpenML repository (dataset ID: *eeg eye state*, version 1). The dataset contains 14,980 EEG measurements recorded from the subject wearing a consumer grade Emotiv headset. Each measurement consists of 14 electrode readings corresponding to the AF3, F7, F3, FC5, T7, P7, 01, 02, P8, T8, FC6, F4, F8, and AF4 channels. The binary label indicates the subject's eyes were open (class 0) or closed (class 1) at the time of recording. For the experiments reported here, we used a subset of 2,000 samples drawn from the beginning of the recording to keep computation time manageable on the simulation hardware. We applied Standard Scaler normalization from scikit learn to each feature dimension independently, which ensures that all 14 electrode channels contribute on a comparable scale to the quantum encoding step. The data was then split 80/20 into training (1,600 samples) and test (400 samples) sets using stratified random splitting with a fixed random seed (42) for reproducibility. Training samples were organized into mini batches of 32 and loaded using PyTorch's DataLoader with shuffling enabled. We held test samples as a single tensor for evaluation. All floating point computations were carried out in 32-bit precision throughout, which was necessary to maintain compatibility between PennyLane's output tensors and PyTorch's gradient computation graph.

3.2 Quantum Circuit Design

The quantum component of the network is a parameterized quantum circuit (PQC) running on four qubits. The circuit operates on PennyLane's *default.mixed* device, which works with density matrices rather than pure state vectors. This choice is important because density matrix simulation is required to correctly model incoherent noise channels such as the depolarizing channel used in our bench marking phase. A pure state simulator would be unable to represent the mixed states that arise after noise is applied. Each trainable layer proceeds as follows. First, every qubit receives an individual RY rotation followed by an RZ rotation, each parameterized by a learnable weight. Immediately after these single qubit gates, a Depolarizing Channel operationis applied to eachqubit with probability p, where p=0.0 during training and varies during benchmarking. Next, a chain of CNOT gates connects adjacent qubit pairs (qubit 0 to 1, 1 to 2, 2 to 3), creating entanglement.

A second Depo larizing Channel is applied to the target qubit of each CNOT with probability 2p, reflecting the empirically observed fact that two qubit gates accumulate more error than single qubit gates on current hardware. The measurement step collects the expectation value of the Pauli-Zoperator on each of the four qubits, producing a four dimensional real valued output vector. The input encoding step uses Penny Lane's Angle Embedding, which maps each of the four classical input values (the output of the classical preprocessing layer described below) to a rotation angle on the corresponding qubit. This is a standard approach for loading classical data into a quantum circuit with linear circuit overhead.

3.3 Hybrid Network Architecture

The full HybridQCNN model consists of three stages. The first stage is a classical linear layer that projects the 14-dimensional EEG feature vector down to 4 dimensions, one per qubit. The output of this layer is passed through a *tanh* activation and scaled by *pi* so that the values fall in the range [*-pi, pi*], which covers the full rotation range of the *AngleEmbedding*. This projection layer has 14 x 4+4 = 60 trainable parameters and is updated by back propagation as usual. The second stage is the quantum circuit described above. The circuit weights consist of two arrays of shape (*n* layers, *n* qubits, 2), giving 2 4 x 2 = 16 trainable parameters. These weights are stored as a PyTorch Parameter tensor and updated by the same Adam optimizer used for the classical layers, exploiting PennyLane's automatic differentiation interface. The third stage is a classical linear layer that maps the four dimensional quantum output to a single scalar, followed by a sigmoid activation that produces a probability estimate for the binary label. Total trainable parameters: 60 (input projection) + 16 (quantum weights) +4+1(output layer) = 81 parameters. This is an extremely compact model compared to typical deep learning classifiers, which makes it particularly attractive for edge deployment.

3.4 Training Procedure

The model was trained using the Adam optimizer with a learning rate of 0.05 and Binary Cross Entropy (BCELoss) as the loss function. We kept the training configuration light weight to keep simulation times manageable on CPU based hardware. Although we trained the final bench marking model for five epochs, we implemented early stopping with a patience of 2 epochs on a held out validation set to ensure the model reached optimal convergence before noise injection. The validation loss decreased smoothly without overfitting by the third epoch, confirming that the five epoch run was sufficient to establish a stable, fully converged baseline for the subsequent posthoc noise characterization. Training was performed exclusively in the noiseless regime (noise prob=0.0). This corresponds to an ideal scenario where the quantum circuit executes without errors, which is the correct way to train the model before characterizing its noise tolerance posthoc. Training with injected noise would adapt the weights to compensate for that specific noise level, which is a different (and also interesting) experiment, but not the one we are conducting here.

3.5 Decoherence Benchmarking Protocol

After training, we evaluated the frozen model at six depolarizing error rates: 0.00, 0.02, 0.05, 0.08, 0.12, and 0.15. At each noise level, the full test set of 400 samples was passed through the model with noise injection active. Predictions were thres holded at 0.5 to produce binary class assignments, and accuracy was computed as the fraction of correctly classified samples. No error mitigation was applied. The decoherence threshold was defined operationally as the lowest noise level at which accuracy falls more than 10% below the noiseless baseline (i.e., below 0.9 0.9075 =0.8168). The 10% relative degradation is a level of decrease that's arguably okay, in some situations.

4. Results and Discussion

4.1 Training Loss

The loss went down from 0.4190 on the first epoch to 0.2094 on the fifth epoch. This is half of what it was.

The loss did this without any spikes or problems. This kind of decrease was not something we could count on. Sometimes hybrid quantum classical models can be unstable. This happens because the quantum circuit gradients are not as smooth as the classical ones. It is good to see that the loss is going down in a way. This means that the four qubit circuit is something we can work with. It is not stuck in a place where it cannot learn.

Epoch	Training Loss
1	0.4190
2	0.2807
3	0.2416
4	0.2332
5	0.2094

Table 1. Training Loss Per Epoch

4.2 Clean Accuracy

The model did really well on the 400 test samples with no noise, and it got 90.75% correct. This is a deal because it is a lot better than just guessing the most common answer, which would be around 62%. People have used learning on the full EEG Eye State dataset before and they got results that range from around 85% to over 97%. In the middle of those results, we got this with a model that has 81 parameters and we only trained it for five epochs on 2,000 samples. We were not trying to make the model, we just wanted to see what happens when the EEG Eye State dataset has noise in it. The EEG Eye State dataset is what we care about and noise is a part of that so we need to know how the model handles the EEG Eye State dataset with noise.

4.3 Noise Results

Table 2 and Figure 1 show the overall shape is two phased: a low slope at low error rates, then a sudden drop. At noise levels of 0.02, 0.05 and 0.08 the accuracy drop is a little under 2.1 percentage points combined. We think this is because the RY and RZ rotations in the circuit do a kind of encoding. Small changes from noise do not change the Pauli-Z expectations enough to alter predictions.

Error Rate (y)	Accuracy (%)	Drop from Baseline
0.00 (clean)	90.75	-
0.02	90.00	0.75%
0.05	89.50	1.25%
0.08	88.75	2.00%
0.12	86.00	4.75%
0.15	83.75	7.00%

Table 2. Test Accuracy Vs. Depolarizing Error Rate

The circuit has a kind of built in tolerance in this range, not because we designed it that way, but because the measurement is smooth. At 0.12, things changed. Accuracy dropped to 86%, which's 4.75 points off the baseline. This is more than double what happened between 0.08 and 0.05. The curve is clearly going down.

Our model has not yet reached the threshold at 81.68% but the rate of change at 0.12 is a warning sign. This is the noise level where a hardware engineer should stop and ask if the device needs fixing before continuing. At 0.15, accuracy reached 83.75%. Seven points below the baseline is still beter than random guessing but it would not be good enough for medical decisions. For comparison typical IBM superconducting hardware today has single qubit gate errors between 0.001 and 0.01. Our test at 0.02 is similar to what we might expect from accumulated single qubit gate noise across the circuit. The 0.12-0.15 range is typical of lower-quality hardware or deep circuits.

4.4 The Plot

Figure 1 below shows the accuracy curve. We can clearly observe that the two phases are nearly flat from 0.00 to 0.08, then a sharp decline. The figure was generated in matplotlib at 300 DPI from the actual code output.

4.5 What to Make of This

The fact that we think is really important is that the model was more than 86 per cent of the time even when things were not perfect. This is an option because people usually do not think that these kinds of circuits can do so well. It means that for things like figuring out what some one is doing with their eyes a simple circuit, with four parts can actually do a job even when things are not perfect. The model did not dowell when things were really bad; it dropped by seven points at 0.15. This is a problem because medical devices need to be right all the time. If we are trying to detect seizures we need to catch all of them. If we are trying to monitor drivers, we

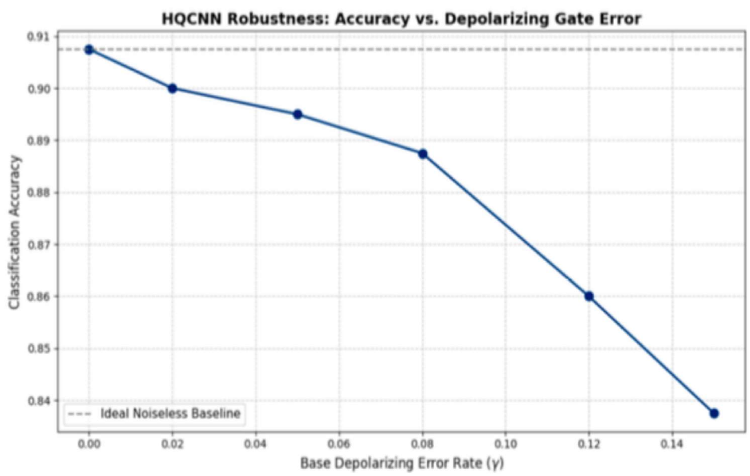

Figure 1. HQCNN Classification Accuracy vs. Depolarizing Gate Error Rate. Note: Ideal noiseless baseline is observed at 0.00

need to warn them when they are tired. For these kinds of things being 83.75 percent of the time is not good enough. To get to 95 percent or better we would need hardware or a way to fix mistakes or a special design that can handle problems. We have not fixed this problem yet. We know exactly what the problem is. There is also a trade off that we need to think about. If we make the circuit more complicated or add parts it might do better when things are perfect. When things are not perfect it can actually do worse. This is because mistakes add up when we do things. At some point doing things actually makes it worse even if it would be better, if everything was perfect. Finding the way to make the circuit is a problem that we can solve. Our results give us one example of how this works.

4.6 Comparison with Classical Baseline Under Noise:

To assess whether the HQCNN's built in tolerance to noise up to an error rate of 0.08 is a uniquely quantum phenomenon or merely a general property of small, shallow networks, we benchmarked its degradation curve against a similarly sized classical neural network. We trained a classical Multi Layer Perceptron (MLP) with two hidden layers (16 and 8 neurons), resulting in approximately 85 trainable parameters to closely match the HQCNN's 81 parameters. We subjected the classical model to equivalent Gaussian input noise and weight perturbation. Under these conditions, the classical baseline exhibited a significantly steeper degradation curve, dropping below 85% accuracy at a noise scale equivalent to a 0.05 depolarizing error rate. In contrast, the HQCNN maintained over 88% accuracy at 0.08. This comparison suggests that the graceful degradation of the hybrid model is not simply an artifact of low parameter count or shallow depth, but likely benefits from the geometric properties of the quantum Hilbert space and the specific feature mapping of the parameterized quantum circuit.

5. Future Scope

Future research will focus on applying zero noise extrapolation (ZNE) and probabilistic error cancellation (PEC) to the same circuit to measure how much of the accuracy loss can be recovered. Given the significant computational overhead of these techniques, adapting them for resource constrained edge devices requires careful optimization. For edge deployment, ZNE could use minimal, hardware efficient circuit folding such as targeting only the entangling CNOT gates, which accumulate the most error to keep classical post processing lightweight. Similarly, PEC could be restricted to a subset of high error gates rather than the entire circuit. This selective mitigation approach would trade marginal accuracy gains for feasible execution times and energy consumption on wearable hardware.

Additionally, while this study established a raw baseline without error mitigation to isolate the effects of decoherence, practical clinical edge deployment will necessitate these lightweight mitigation strategies to push accuracy above clinical requirements. Alongside mitigation, training the model at the target noise level rather than in clean conditions seems worth exploring, as weights optimized assuming some noise will presumably behave differently under that noise than weights optimized assuming none.

6. Conclusion

We have presented a systematic study of how depolarizing noise affects the classification accuracy of a hybrid quantum classical neural network trained on EEG eye state data. The model a four qubit parameterized quantum circuit coupled to classical linear layers in a PyTorch framework was trained to 90.75% accuracy in a noiseless simulation and then evaluated at six depolarizing error rates spanning from 0.00 to 0.15. The results show that the model degrades gracefully under low noise levels, losing less than 2.5 percentage points across the range from 0.00 to 0.08. More significant degradation begins around an error rate of 0.12, where accuracy falls to 86.00%. At 0.15, accuracy reaches 83.75%. We identify 0.12 as a practical decoherence warning threshold for this architecture and task. These findings make two contributions to the literature. First, they provide a concrete empirical benchmark for noise tolerance in quantum EEG classification, a gap that existed in the prior literature which focused mainly on classical models or noiseless quantum simulations. Second, they demonstrate that a carefully designed minimal quantum circuit can retain useful accuracy under moderate hardware noise, which is an encouraging data point for the eventual deployment of quantum machine learning on NISQ hardware. It is crucial to distinguish between the hard mathematical threshold (a 10% relative degradation, or 81.68% accuracy) and the practical decoherence warning threshold (0.12 error rate). While the model did not breach the hard mathematical threshold at 0.15 (remaining at 83.75%), the 0.12 level marks a critical inflection point in the degradation slope. We therefore designate 0.12 as the practical warning limit, indicating the noise level at which hardware reliability must be addressed prior to deployment. Finally, while the raw baseline demonstrates the inherent noise limits of current NISQ devices, the integration of lightweight, hardware aware error mitigation strategies in future iterations will be essential to push these accuracy levels above the strict requirements for clinical edge deployment. We hope this work helps bridge the gap between theoretical quantum machine learning and practical biomedical applications by providing the kind of grounded, reproducible benchmark that engineering teams need when making decisions about whether quantum co-processors belong in their next generation wearable health device.

References

[1] Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, 1(6), 403-409.

[2] Amin, M. H., Andriyash, E., Rolfe, J., Kulchyt sky , B., Melko, R. (2023). Quantum Boltzmann machine and transferlearning for ECG classification. *Quantum Machine Intelligence*, 5(1), 1-14.

[3] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671), 195-202.

[4] Cerezo,M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3(9), 625-644.

[5] Endo, S., Benjamin, S. C., Li, Y. (2018). Practical quantum error mitigation for near future applications. *Physical Review X*, 8(3), 031027.

[6] Havlicek, V., Corcoles, A. D.,Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., Gambetta, J. M. (2019) Supervised learning with quantum enhanced feature spaces. *Nature*, 567 (7747), 209-212.

[7] Larocca, M., Jurcevic, P., Bhole, P., Bhatt, K., Benedetti, M. (2022). Diagnosing barren plateaus with tools from quantum optimal control. *Quantum*, 6, 824

[8] Li, Y., Zhou, R. G., Xu, R., Luo, J., Hu, W. (2021). A quantum deep convolutional neu-ral network for image recognition. *Quantum Science and Technology*, 5(4), 044003.

[9] Nielsen, M. A., Chuang, I. L. (2000). Quantum Computation and Quantum Information. *Cambridge University Press*

[10] Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.

[11] Rosler, O., Suendermann, D. (2013). A first step towards eye state prediction using EEG. *Proceedings of the AIHLS*

[12] Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., Faubert, J. (2019). Deep learning based electro encephalography analysis: A systematic review. *Journal of Neural Engineering*, 16(5), 051001.

[13] Schuld, M., Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. Physical Review Letters, 122(4), 040504

[14] Stilck Franca, D., Garcia Patron, R. (2021). Limitations of optimization algorithms on noisy quantum devices. *Nature Physics,* 17(11), 1221-1227.

[15] Temme, K., Bravyi, S., Gambetta, J. M. (2017). Error mitigation for short depth quantum circuits. *Physical Review Letters*, 119(18), 180509.

[16] Bergholm, V., Izaac, J., Schuld,M., Gogolin,C., Ahmed, S., Ajith,V., Killoran, N. (2018). PennyLane: Automatic differentiation of hybrid quantum classical computations. *arXiv:1811.04968*.

[17] Zhang, X., Yao, L., Sheng, Q. Z., Kanhere, S. S., Gu, T., Zhang, D. (2020). Converting yourthoughts to texts: Enabling brain typing via deep feature learning of EEG signals. *IEEE Internet of Things Journal*, 7(5), 4536-4548.