ROBUST AND EFFICIENT UNCERTAINTY AWARE BIOSIGNAL CLASSIFICATION VIA EARLY EXIT ENSEMBLES

Alexander Campbell[*1,2], *Lorena Qendro*[*1], *Pietro Liò*[1], *Cecilia Mascolo*[1]

[1]University of Cambridge, [2]The Alan Turing Institute

ABSTRACT

Ensembles of deep learning models can be used for estimating predictive uncertainty. Existing ensemble approaches, however, introduce a high computational and memory cost limiting their applicability to real-time biosignal applications (e.g. ECG, EEG). To address these issues, we propose early exit ensembles (EEEs) for estimating predictive uncertainty via an implicit ensemble of early exits. In particular, EEEs are a collection of weight sharing sub-networks created by adding exit branches to any backbone neural network architecture. Empirical evaluation of EEEs demonstrates strong performance in accuracy and uncertainty metrics as well as computation gain highlighting the benefit of combining multiple structurally diverse models that can be jointly trained. Compared to state-of-the-art baselines (with an ensemble size of 5), EEEs can improve uncertainty metrics up to $2\times$ while providing test-time speed-up and memory reduction of approx. $5\times$. Additionally, EEEs can improve accuracy up to 3.8 percentage points compared to single model baselines.

***Index Terms*—** Uncertainty, Deep Learning, Early Exit

1. INTRODUCTION

Ensembles of deep learning models can improve predictive performance by combining the output of multiple models [1, 2]. The majority of approaches, however, focus on accuracy while neglecting predictive uncertainty. For biosignal classification, uncertainty quantification is critical since the input distribution is often shifted from the training distribution due to different hardware and/or data collection protocols [3]. In such scenarios, a model with well-calibrated uncertainty can indicate if a prediction should be trusted [4].

Recently, ensemble deep learning techniques have been shown to provide reliable predictive uncertainty quantification. These approaches can be broadly divided into implicit vs. explicit ensembles. Monte Carlo dropout [5] creates an implicit ensemble of networks by approximately sampling a weight distribution during inference. On the other hand, deep ensembles [6] and hyper-deep ensembles [7] are explicit ensembles composed of models of the same architecture independently trained with different weight initialization and hyperparameters, respectively. These aforementioned techniques, however, have limited use for real-time biosignal tasks due to their computational and memory overhead. In particular, Monte Carlo dropout requires multiple forward passes to create an ensemble which translates into higher latency and computational cost. On the other hand, for deep and hyper-deep ensembles, not only does the memory footprint scale linearly with the ensemble size, but also an increased computational burden is incurred from loading and running multiple models.

To address the computational and memory bottleneck of the previous techniques for uncertainty quantification, we propose early exit ensembles (EEEs) via a novel interpretation of early exit neural networks [8, 9]. In our approach, EEEs form an implicit ensemble of models from which predictive uncertainty can be quantified. Specifically, EEEs are a collection of weight sharing sub-networks which can be created by adding exit branches to any backbone deep learning architecture. We demonstrate that EEEs are not only easy-to-implement but also perform optimally in terms of accuracy, uncertainty quantification, run-time, and memory on multiple model-dataset combinations. Furthermore, we focus our empirical analysis on the task of biosignal classification as it remains a highly unexplored area in uncertainty estimation literature to date. Our code is publicly available at https://github.com/ajrcampbell/early-exit-ensembles.

2. METHODS

Consider a classification problem where $\mathbf{x} \in \mathbb{R}^D$ denotes a D-dimensional input and $y \in \{1, \dots, C\}$ a corresponding discrete target taking one of C classes. We aim to learn a neural network (NN) $f_\theta(\cdot)$ that can model the predictive distribution $p_\theta(y|\mathbf{x})$ over ground truth labels, given model parameters θ. By definition, a NN consists of blocks of differential operations (e.g., convolution). We assume therefore $f_\theta(\cdot)$ can be decomposed into B blocks such that $f_\theta(\mathbf{x}) = (f^{(B)} \circ f^{(B-1)} \circ \dots \circ f^{(1)})(\mathbf{x})$ where $(f^{(i)} \circ f^{(j)})(\mathbf{x}) = f_{\theta_i}(f_{\theta_j}(\mathbf{x}))$ denotes function composition for $i \neq j$ and $\theta = \cup_{i=1}^{B} \theta_i$. Let $\mathbf{h}^{(i)} \in \mathbb{R}^{K_i \times D_i}$ denote the intermediary output of the i-th block having K_i features of dimension $D_i \leq D$ such that $\mathbf{h}^{(i)} = f_{\theta_i}(\mathbf{h}^{(i-1)})$ for $1 \leq i \leq B-1$, and $\mathbf{h}^{(0)} = \mathbf{x}$.

[*]Equal contribution

2.1. Early Exit Ensemble

We define an early exit block as a NN $g_{\phi_i}(\cdot)$ which takes as input the intermediary output $\mathbf{h}^{(i)}$ from the i-th block of $f_\theta(\cdot)$, henceforth referred to as the backbone. We let each exit block learn a predictive distribution $p_{\phi_i}(y|\mathbf{x}) = \sigma(g_{\phi_i}(\mathbf{h}^{(i)}))$ where $\sigma(\cdot)$ is the softmax transform. As such, any NN is able to output a set $\mathcal{M} = \{p_{\phi_1}(y|\mathbf{x}), \ldots, p_{\phi_{B-1}}(y|\mathbf{x}), p_\theta(y|\mathbf{x})\}$ which represents an EEE. The ensemble $\mathcal{M}$ contains up to $B-1$ distributions from early exits blocks, in addition to the standard output from its final block. As such, ensemble size $|\mathcal{M}| = B$.

To train an EEE, we optimize a weighted sum of each exits' individual predictive loss. This procedure allows the training of the whole ensemble jointly. More formally:

$$\mathcal{L} = L_{CE}(y, f_\theta(y|\mathbf{x})) + \sum_{i=1}^{B-1} \alpha_i L_{CE}(y, g_{\phi_i}(y|\mathbf{x}))$$

where $L_{CE}(\cdot,\cdot)$ is the cross-entropy loss function and $\alpha_i \in [0, 1]$ is a weight hyperparameter corresponding to the relative importance of each exit.

During inference, a single forward pass of a NN with early exits produces an ensemble $\mathcal{M}$ of predictions. The overall prediction from $\mathcal{M}$ can be computed as the mean of a categorical distribution obtained from averaging the predictions from the individual exits:

$$p_{\theta_{\mathcal{M}}}(y|\mathbf{x}) \approx \frac{1}{|\mathcal{M}|}\big(p_\theta(y|\mathbf{x}) + \sum_{i=1}^{B-1} p_{\phi_i}(y|\mathbf{x})\big).$$

Compared to a single model prediction, an ensemble provides more information such as variance, entropy and disagreement (as measured by Kullback-Leibler (KL) divergence), that can be exploited for better-calibrated predictive probabilities during both training and inference [10, 11].

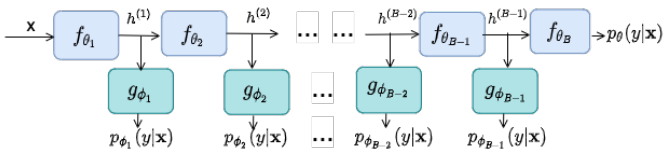


Fig. 1: Representation of an early exit ensemble.

2.2. Exit Block Architecture

Exits from earlier blocks inherit intermediary outputs with weaker representational capacity, which negatively impacts ensemble accuracy. To address this issue, we design a conditional architecture for the i-th exit block as follows:

$$g_{\phi_i}(\mathbf{h}^{(i)}) = \begin{cases} \mathbf{W}_2^{(i)}\rho(\mathbf{W}_1^{(i)}s(\mathbf{h}^{(i)}) + \mathbf{b}_1^{(i)}) + \mathbf{b}_2^{(i)} & \gamma > 0 \\ \mathbf{W}_3^{(i)}s(\mathbf{h}^{(i)}) + \mathbf{b}_3^{(i)} & \gamma = 0 \end{cases}$$

where $s(\cdot)$ denotes global average pooling, $\rho(\cdot)$ is an activation function, $\mathbf{W}_1^{(i)} \in \mathbb{R}^{K_\gamma \times K_i}$, $\mathbf{W}_2^{(i)} \in \mathbb{R}^{C \times K_\gamma}$, $\mathbf{W}_3^{(i)} \in \mathbb{R}^{C \times K_i}$ and $\mathbf{b}_1^{(i)} \in \mathbb{R}^{K_\gamma}$, $\mathbf{b}_2^{(i)}, \mathbf{b}_3^{(i)} \in \mathbb{R}^C$ are weights and biases of linear layers respectively. The hyperparameter $\gamma \geq 0$ is a learning capacity factor used to increase the number of features from K_i to K_γ of the i-th intermediary output such that $K_\gamma = (\sqrt{1+\gamma})^{B-i}$ for $1 \leq i \leq B-1$ where K_B is the number of features in the last block defined by the backbone. Intuitively, when $\gamma > 0$ the number of features in each exit block is inversely proportional to the exit point i.e. earlier exits use additional parameters to learn more complex relations between features.

2.3. Exit Placement

In practice, the number of exit blocks is determined by the backbone architecture as well as computational cost [12] and quality of the provided uncertainty estimates [4]. Therefore the ensemble size $|\mathcal{M}|$ is a hyperparameter bounded above by B. As such, there are a combinatorial choice of exist points and therefore ensemble arrangements. To limit the search space, we introduce exit placement strategies in Table 1.

Strategy	Exit after
Block-wise	Every block.
Pareto	Blocks closest to 20% and 80% of total FLOPs.
Computation	Blocks closest to {15, 30, 45, 60, 75, 90}% of total FLOPs.
Residual	Residual blocks.
Last-k	Last k blocks.
Semantic	Last block grouped by number and size of feature maps.

Table 1: Exit placement strategies for any backbone architecture. FLOPs: floating point operations.

For example, given a ResNet18 backbone (with blocks defined as convolution, batch normalization, and activation), the $Semantic$ strategy places exit blocks after each of the last blocks of kernel size and number features $(3, 64)$, $(3, 128)$, $(3, 256)$, and $(3, 512)$ resulting in size $|\mathcal{M}| = 5$.

2.4. Computational Cost

Table 2 compares computational cost of EEEs vs. an ensemble of independent models and a single model (assuming the same backbone). The only memory overhead introduced by EEEs are the parameters from exit blocks $\phi = \cup_{i=1}^{|\mathcal{M}|-1}\phi_i$. Since in general $\phi << \theta$ where $\theta = \cup_{i=1}^{B}\theta_i$, EEEs achieve computational and memory gains due to weight sharing.

Ensemble	Size	FLOPs	Compute
Single	θ	F	τ
Independent	$\theta * \lvert\mathcal{M}\rvert$	$F * \lvert\mathcal{M}\rvert$	$\tau * \lvert\mathcal{M}\rvert$
Early exit	$\theta + \phi$	$F + F_\phi$	$\tau + \tau_\phi$

Table 2: Computational and memory cost. F: FLOPs. τ: compute time. F_ϕ and τ_ϕ are FLOPs (floating-point operations) and compute time for all exit blocks respectively.

3. EXPERIMENTS

Datasets & architectures. We evaluate our approach on the task of biometric signal classification using three datasets: ECG heart attack (ECG) [13], EEG epileptic seizure (EEG-S) [14], and EEG artifacts (EEG-A) [15] where only eye-movement artifacts are considered. All datasets are split into $80\%/10\%/10\%$ train/validation/test maintaining class proportions. Each dataset is paired with a different architecture: FCNet [16] for ECG, ResNet18 [17] for EEG-S, and VGG16 [18] for EEG-A. For ResNet18 and VGG16, convolutional layers are made 1-dimensional following previous work on biosignal classification [19, 20].

Baselines. We compare EEEs (Early Exit) against its backbone architecture (Backbone), Monte Carlo dropout (MCDrop) [5], deep ensembles (Deep) [6] and depth ensembles (Depth). Depth is introduced as an explicit ensemble where each model ranges from shallow to deep based on the same backbone architecture. Following findings on optimal size for well-calibrated uncertainty [4], we set ensemble size to 5 for all models. For fairness of comparison, depth is determined by the placement of exit points from Early Exit. Dropout layers in MCDrop are similarly placed at exit points.

Metrics. Performance is evaluated using class weighted F1, negative log-likelihood (NLL), Brier score (BS), and expected calibration error (ECE) (see [4] for an overview). NLL measures how likely it is to observe the test data given each trained model, BS measures the accuracy of predicted probabilities, and ECE measures model calibration as the expected difference between accuracy and predicted confidence.

Hyperparameters. All models are trained using the Adam optimizer [21] and an optimally tuned learning rate, batch size, and epochs: FCNet ($1e^{-2}$, 200, 250), ResNet18 ($1e^{-3}$, 200, 200), VGG16 ($1e^{-4}$, 200, 200). For MCDrop, the optimal dropout rate is 0.2. All results for Early Exit are for a loss with α_i =1 as well as a learning capacity factor and exit strategy optimally tuned as follows: FCNet (γ=0.0, *Block-wise*), ResNet18 (γ=0.2, *Semantic*), and VGG16 (γ=0.5, *Semantic*). To prevent overfitting, early-stopping is used with validation accuracy and a patience of 5.

3.1. Classification and predictive uncertainty

Table 3 summarizes accuracy and uncertainty results. In terms of accuracy (as measured by F1 score), Early Exit performs best in two out of three datasets (ECG and EEG-A) compared to the best baseline Deep. Across all backbone architectures, Early Exit improves accuracy by up to 3.8 percentage points. These findings reflect the variance reducing effect of averaging a set of diverse models with individually high variance and low bias [22] (see Section 3.4). With regards to uncertainty, Early Exit outperforms all baselines (as measured by NLL, ECE, and BS). The biggest gain is reflected in FCNet where Early Exit improves on uncertainty metrics by up to $2\times$ compared to baselines. Finally, compared to the current state-of-the-art Deep, Early Exit improves calibration as measured by ECE for ResNet18 (0.028 vs 0.027) and VGG16 (0.039 vs 0.033) representing a 3.8% and 15.4% decrease, respectively. In general, this should translate into Early Exit displaying greater/lesser uncertainty under out/in-distribution data shifts (see Section 3.2).

[1]Running on a 2.7 GHz Intel Core i7 CPU.

	F1 (↑)	NLL (↓)	ECE (↓)	BS (↓)
FCNet	0.983 (.010)	0.059 (.031)	0.009 (.005)	0.026 (.015)
- MCDrop	0.987 (.002)	0.043 (.019)	0.011 (.004)	0.019 (.004)
- Depth	0.989 (.007)	0.036 (.008)	0.017 (.007)	0.018 (.006)
- Deep	0.989 (.003)	0.045 (.020)	0.014 (.007)	0.018 (.005)
- Early exit	**0.992 (.005)**	**0.024 (.008)**	**0.007 (.001)**	**0.009 (.003)**
ResNet18	0.847 (.012)	0.432 (.022)	0.081 (.007)	0.233 (.018)
- MCDrop	0.844 (.007)	0.362 (.012)	0.045 (.005)	0.216 (.011)
- Depth	0.861 (.011)	0.318 (.028)	0.028 (.003)	0.194 (.012)
- Deep	**0.866 (.009)**	0.316 (.024)	0.028 (.004)	0.189 (.011)
- Early exit	0.865 (.002)	**0.306 (.013)**	**0.027 (.006)**	**0.189 (.005)**
VGG16	0.809 (.011)	0.589 (.040)	0.109 (.020)	0.308 (.015)
- MCDrop	0.822 (.011)	0.574 (.051)	0.093 (.012)	0.279 (.014)
- Depth	0.821 (.023)	0.438 (.018)	0.057 (.009)	0.275 (.010)
- Deep	0.838 (.010)	0.400 (.004)	0.039 (.004)	0.239 (.007)
- Early exit	**0.847 (.003)**	**0.385 (.005)**	**0.033 (.010)**	**0.236 (.003)**

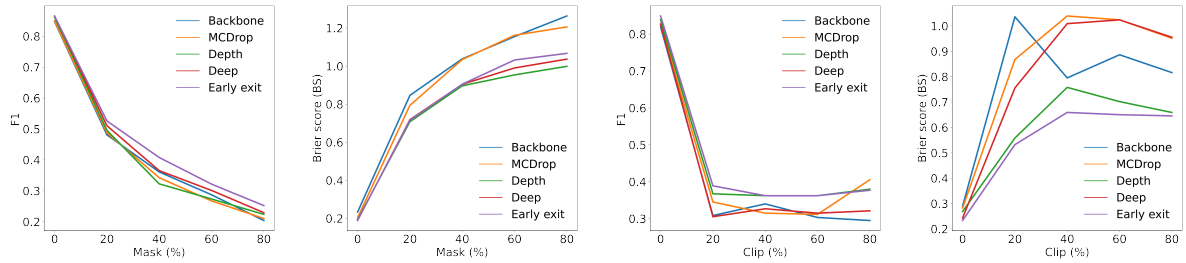
Table 3: Classification and uncertainty results. Entries are mean and standard deviation over 3 random splits of test data. Best results are indicated in bold.

			MCDrop	Depth	Deep	Early exit
FCNet	**Size**	(↓)	**0.20**	<u>0.70</u>	1.10	**0.20**
	FLOPs	(↓)	0.30	<u>0.19</u>	0.30	**0.06**
	Time	(↓)	6.57	<u>4.90</u>	6.65	**1.93**
ResNet18	**Size**	(↓)	**3.80**	8.90	19.20	<u>4.40</u>
	FLOPs	(↓)	0.33	<u>0.18</u>	0.33	**0.07**
	Time	(↓)	21.10	<u>11.50</u>	21.40	**4.20**
VGG16	**Size**	(↓)	**23.80**	30.10	119.00	<u>25.80</u>
	FLOPs	(↓)	11.60	<u>6.30</u>	11.60	**2.30**
	Time	(↓)	52.54	<u>31.57</u>	62.87	**11.53**

Table 4: Memory and efficiency results. Size: number of parameters (millions). FLOPs: number of floating point operations (Giga). Time: average inference time over test dataset (milliseconds)[1]. Best/second best results are indicated by bold/underline.

3.2. Calibration on in-distribution shifts

Figure 2 displays the effect of in-distribution shifts on EEG-S and EEG-A test data for ResNet18 and VGG16, respectively. We apply two types of shifts: signal masking and amplitude clipping. Signal masking represents missing EEG signal caused by electrode movement or temporary malfunction. Signal clipping, instead, represents amplifier saturation caused by excess voltage. For all models, accuracy decreases



(a) ResNet18 F1 score (↑) (b) ResNet18 Brier score (↓) (c) VGG16 F1 score (↑) (d) VGG16 Brier score (↓)

Fig. 2: In-distribution shifts applied to $[0\%, 20\%, 40\%, 60\%, 80\%]$ of test data for ResNet18 (masking) and VGG16 (clipping). A well-calibrated model shows higher F1 and lower Brier score across all shift percentages.

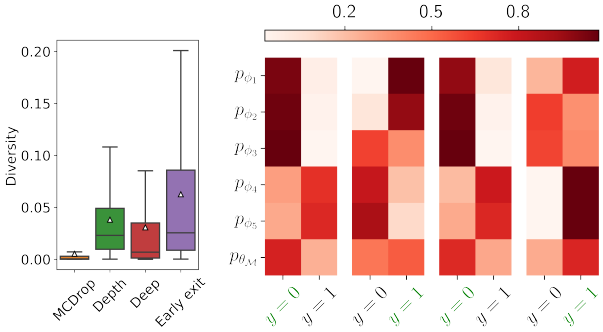
as each shift intensifies. However, Early Exit displays greater robustness by maintaining a higher F1 score across all intensities for both shift types compared to baselines. In terms of the accuracy of predicted probabilities (as measured by BS), Early Exit performs better on clipping for VGG16 while following a similar trend to Deep and Depth for ResNet18 up to shift intensity of 40%.

3.3. Efficiency analysis

Table 4 summarizes memory and efficiency analysis for all models. As expected, the memory footprint (as measured by size) of Deep is the highest since it consists of 5 independently trained models. The most memory efficient model is MCDrop, however, the FLOPs required for uncertainty estimation is approximately $5\times$ higher than Early Exit due to sampling at inference time. Dealing with 5 models (Deep and Depth) or performing 5 samples (MCDrop) translates into higher inference time compared to Early Exit. Overall, Early Exit presents the best trade-off between memory (max 16% increase), inference time (approx. $5\times$ less), and number of FLOPs (approx. $5\times$ less) since EEEs can produce all ensemble members' predictions in a single forward pass. Therefore, EEEs are better suited to a wider range of real-world applications in need of efficient uncertainty quantification.

3.4. Diversity analysis

In Figure 3a we visualize diversity computed as the KL divergence between each ensemble member prediction and the average ensemble prediction [23]. Given the comparable median diversity of Early Exit and Depth (approx. 0.024), it is clear that varying network structure leads to higher predictive diversity. Overall, we conclude that Early Exit is the best preforming technique as it has a greater range of disagreement at a much lower computational cost, since it can provide multiple predictions in a single forward pass. The heat maps of predictive confidence in Figure 3b (4 samples from EEG-A) further highlight the importance of ensemble diversity. Averaging the predicted probabilities of each member of Early Exit results in a correct prediction (highlighted in green) even though some individual members are wrong. This finding is in line with previous work suggesting that a good performing ensemble should be both accurate and diverse [22].



(a) Diversity (↑) (b) Early Exit predictive confidence (↑)

Fig. 3: Visualizing (a) diversity, and (b) predictive confidence (Early Exit, 4 samples) for VGG16. Ground truth in green.

4. CONCLUSION

We propose early exit ensembles (EEEs), an efficient and easy-to-implement implicit ensemble technique for uncertainty quantification in deep learning biosignal classification tasks. Our approach achieves remarkable performance improvements over previous state-of-the-art ensemble deep learning techniques presenting the best trade-off among accuracy, uncertainty quantification, run-time, and memory on a wide variety of datasets and architectures. The strong performance of EEEs highlights the importance of structural diversity when building a well performing ensemble. We believe that the simplicity of our framework, combined with its strong transferability across architectures and datasets, positions it as a *de facto* baseline for future work on uncertainty quantification in biosignal classification.

5. ACKNOWLEDGMENTS

This work is supported by Nokia Bell Labs through their donation for the Centre of Mobile, Wearable Systems and Augmented Intelligence, ERC Project 833296 (EAR) as well as The Alan Turing Institute under the EPSRC grant EP/N510129/1. We thank the UCI Machine Learning Repository, UCR Time Series Classification Archive, and the Temple University EEG Corpus for providing the datasets.

6. REFERENCES

[1] Abdolrahman Peimankar and Sadasivan Puthusserypady, "An ensemble of deep recurrent neural networks for p-wave detection in electrocardiogram," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 1284–1288.

[2] Xiao Zheng, Wanzhong Chen, Yang You, Yun Jiang, Mingyang Li, and Tao Zhang, "Ensemble deep learning for automated visual classification using eeg signals," *Pattern Recognition*, vol. 102, pp. 107147, 2020.

[3] Gustav Mårtensson, Daniel Ferreira, Tobias Granberg, Lena Cavallin, Ketil Oppedal, Alessandro Padovani, Irena Rektorova, Laura Bonanni, Matteo Pardini, Milica G Kramberger, et al., "The reliability of a deep learning model in clinical out-of-distribution mri data: a multicohort study," *Medical Image Analysis*, vol. 66, pp. 101714, 2020.

[4] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *arXiv preprint arXiv:1906.02530*, 2019.

[5] Yarin Gal and Zoubin Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[6] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[7] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton, "Hyperparameter ensembles for robustness and uncertainty quantification," *arXiv preprint arXiv:2006.13570*, 2020.

[8] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2464–2469.

[9] Alessandro Montanari, Manuja Sharma, Dainius Jenkus, Mohammed Alloulah, Lorena Qendro, and Fahim Kawsar, "eperceptive: energy reactive embedded intelligence for batteryless sensors," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 382–394.

[10] Lorena Qendro, Alexander Campbell, Pietro Lio, and Cecilia Mascolo, "Early exit ensembles for uncertainty quantification," in *Machine Learning for Health*. PMLR, 2021, pp. 181–195.

[11] Lorena Qendro, Alexander Campbell, Pietro Liò, and Cecilia Mascolo, "High frequency eeg artifact detection with uncertainty via early exit paradigm," *arXiv preprint arXiv:2107.10746*, 2021.

[12] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras, "Shallow-deep networks: Understanding and mitigating network overthinking," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3301–3310.

[13] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.

[14] Dheeru Dua and Casey Graff, "Uci machine learning repository," 2017.

[15] Ahmed Hamid, Katherine Gagliano, Safwanur Rahman, Nikita Tulin, Vincent Tchiong, Iyad Obeid, and Joseph Picone, "The temple university artifact corpus: An annotated corpus of eeg artifacts," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE, 2020, pp. 1–4.

[16] Zhiguang Wang, Weizhong Yan, and Tim Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] Paschalis Bizopoulos, George I Lambrou, and Dimitrios Koutsouris, "Signal2image modules in deep neural networks for eeg classification," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 702–705.

[20] Kit Hwa Cheah, Humaira Nisar, Vooi Voon Yap, Chen-Yi Lee, and GR Sinha, "Optimizing residual networks and vgg for classification of eeg signals: Identifying ideal channels for emotion recognition," *Journal of Healthcare Engineering*, vol. 2021, 2021.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Michael P Perrone and Leon N Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," Tech. Rep., BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS, 1992.

[23] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan, "Deep ensembles: A loss landscape perspective," *arXiv preprint arXiv:1912.02757*, 2019.