國立臺灣大學共同教育中心國際學院
智慧醫療與健康資訊碩士學位學程

碩士論文

Master's Program in Smart Medicine and Health Informatics

Center of General Education, International College

National Taiwan University

Master's Thesis

注意力深度學習方法應用於時間序列腦電波圖針對心跳停止後腦神經損傷的預後預測

An Attention-Based Deep Learning Approach of Using Time-Series EEG for Predicting Neurological Outcomes in Cardiac Arrest

曾世傑

Jefferson Sy Dionisio

指導教授: 林澤 博士

Advisor: Che Lin, Ph.D.

中華民國 113 年 7 月

July 2024

doi:10.6342/NTU202403906

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

注意力深度學習方法應用於時間序列腦電波圖針對心跳停止後腦神經損傷的預後預測

An Attention-Based Deep Learning Approach of Using Time-Series EEG for Predicting Neurological Outcomes in Cardiac Arrest

The undersigned, appointed by the Department / Institute of Master's Program in Smart Medicine and Health Informatics on 30 (date) July (month) 2024 (year) have examined a Master's thesis entitled above-presented by Jefferson Sy Dionisio (name) R11H45004 (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

林澤	林宛爭	劉子毓
（指導教授 Advisor）		
曹昱	田中聡久	

系主任/所長 Director: 國際學位智慧健康資訊學程主任 林澤

誌謝

我想對我的指導教授林澤教授表達由衷的感謝。謝謝您在研究所期間一直相信我的潛力，並始終支持我的學術成長。很感謝實驗室同伴們的互動和建議，通過我們實驗室 iDSSP，使我每天都有所進步和學習。我也要感謝擔任本論文審查委員的林亮宇教授、田中聰久教授、劉子毓教授和曹昱教授。您們寶貴的指導和反饋對確保我能擁有高質量的研究。

此外，我還要感謝來自國立臺灣大學的教授和同學們，他們在學術和生活方面教會了我寶貴的課程。我在這裡獲得的知識和經驗對我的個人成長和職業發展至關重要。在臺灣的留學生活雖然充滿挑戰，但我很幸運遇到了支持我的朋友，他們幫助我把臺灣變成了我的第二個家。

最後，我要向在菲律賓的家人和朋友們致以衷心的感謝，感謝你們在這段旅程開始以來一直不懈地支持我。你們的不懈支持是我在這段經歷中最大的力量來源。

doi:10.6342/NTU202403906

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Che Lin, for believing in my potential from the very beginning of my graduate school journey and for consistently supporting my academic growth. Through our lab, iDSSP, I have had the opportunity to grow and learn new things daily, thanks to the interactions and support from my lab mates. I am also grateful to the professors who served as committee members for this thesis, Prof. Lian-Yu Lin, Prof. Toshihisa Tanaka, Prof. Joyce Liu, and Prof. Yu Tsao. Their invaluable guidance and feedback have been instrumental in ensuring the highest quality of my work.

Additionally, I wish to acknowledge my professors and peers from NTU, who have taught me priceless lessons in both academics and life. The knowledge and experiences I gained here have been essential to my personal and professional development. Studying abroad in Taiwan has been challenging, but I have been fortunate to meet supportive friends who helped make Taiwan my second home.

Finally, I extend my heartfelt thanks to my family and friends in the Philippines, who have continuously supported me since the beginning of this journey. Their unwavering support has been my greatest strength throughout this experience.

doi:10.6342/NTU202403906

摘要

突發性心臟驟停（SCA）患者通常因缺氧時間過長而陷入昏迷，醫師需提供神經系統預後，協助臨床決策。本研究旨在利用早期腦電圖（EEG）數據訓練 Transformer 模型，預測 SCA 昏迷患者的神經系統預後。Transformer 模型利用自注意力機制從長序列中學習模式。我們利用完整的小時級 EEG 序列，將其分割爲 5 分鐘的時段，使模型能夠捕捉長距離的時間序列模式。通過將每個 EEG 序列視爲訓練樣本，我們增加了數據樣本量，提高了模型學習特定記錄模式的能力。預測結果按患者進行了整合評估。專注於 EEG 數據的模型展現出了良好的預測性能，在保留測試集上的 AUROC 爲 0.82，AUPRC 爲 0.90，在外部測試集上的 AUROC 爲 0.73，AUPRC 爲 0.93。本研究凸顯了注意力機制在識別 EEG 序列中時間模式方面的潛力，提升了對 SCA 患者預後的能力。

關鍵字：腦電圖分類、心臟驟停、多頭注意力機制、結果預測、時間序列數據、變壓器

doi:10.6342/NTU202403906

Abstract

Surviving sudden cardiac arrest (SCA) patients often remain in a coma due to a prolonged lack of oxygen, requiring physicians to provide prognoses on neurological outcomes to aid in clinical decisions. This study aims to predict neurological outcomes in SCA coma patients using early electroencephalogram (EEG) data to train a Transformer model, which leverages self-attention to learn patterns from lengthy sequences. We utilized full hours of EEG sequences, subdividing them into 5-minute epochs, allowing the model to capture long-distance time series patterns. By treating each individual EEG sequence as a training sample, we increased our data sample size and improved the model's ability to learn recording-specific patterns. Predictions were aggregated for patient-wise evaluation. Focusing exclusively on EEG data, our model demonstrated promising predictive performance, with an AUROC of 0.82 and an AUPRC of 0.90 on the holdout test set, and an AUROC of 0.73 and an AUPRC of 0.93 on an external test set. This study underscores the potential of attention mechanisms to discern temporal patterns in EEG

doi:10.6342/NTU202403906

sequences, enhancing SCA patient prognosis.

Keywords: EEG classification, cardiac arrest, multi-head attention, outcome prediction, time series data, Transformer

doi:10.6342/NTU202403906

Contents

doi:10.6342/NTU202403906

doi:10.6342/NTU202403906

doi:10.6342/NTU202403906

List of Figures

doi:10.6342/NTU202403906

doi:10.6342/NTU202403906

doi:10.6342/NTU202403906

List of Tables

doi:10.6342/NTU202403906

Abbreviation

AUPRC	Area under Precision-Recall Curve
AUROC	Area under Receiver Operating Curve
Bi-LSTM	Bidirectional Long Short Term Memory
CPC	Cerebral Performance Category
ECG	Electrocardiogram
EEG	Electroencephalogram
FPR	False Positive Rate
FC	Fully Connected
FF	Feedforward
FN	False Negative
FP	False Positive
GAP	Global Average Pooling

doi:10.6342/NTU202403906

GMP	Global Max Pooling
ICA	Independent Component Analysis
ICU	Intensive Care Unit
I-CARE	International Cardiac Arrest Research Consortium
IQR	Interquartile Range
MHA	Multi-head Attention
NTUH	National Taiwan University Hospital
OHCA	Out-of-Hospital Cardiac Arrest
PCA	Principal Component Analysis
PRC	Precision-Recall Curve
PW	Patient-Wise
QEEG	Quantitative Electroencephalogram
REF	Reference Channels
ReLU	Rectified Linear Unit
ROC	Receiver Operating Curve
ROSC	Return of Spontaneous Circulation
RW	Recording-wise

doi:10.6342/NTU202403906

SCA	Sudden Cardiac Arrest
TN	True Negative
TP	True Positive
TPR	True Positive Rate
TTM	Targeted Temperature Management

doi:10.6342/NTU202403906

Chapter 1

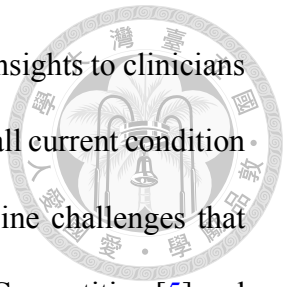
Introduction

1.1 Motivation

Recently, statistics have shown a steady increase in sudden cardiac arrest (SCA) cases due to increasing unhealthy lifestyles across many countries [1]. Patients who experience SCA often arrive in the intensive care units (ICU) of hospitals several minutes or hours after their initial cardiac arrest. Oftentimes, these patients remain in the coma state due to prolonged lack of oxygen, which causes hypoxic-ischemic brain injuries [2,3]. The future outcomes of coma patients remain uncertain, and physicians have to inform patients' guardians on whether the patient is predicted to recover from the coma or to remain in a vegetative or neurologically impaired state. This patient prognosis is extremely important as bad prognoses might result in the discontinuation of life support. Physicians, in turn, are tasked by clinical protocols to only provide patient prognoses on neurological outcomes of patients 72 hours after the return of spontaneous circulation (ROSC) [4].

In the clinical setting, electroencephalogram (EEG) has become an important signal biomarker extracted to aid medical experts in studying the underlying brain conditions of

doi:10.6342/NTU202403906



their patients. These signals, when interpreted visually, may provide insights to clinicians when predicting future outcomes for patients since they show the overall current condition of the brain. In recent years, we have seen an upward trend in online challenges that involve the usage of open-source EEG datasets, such as the 2003 BCI Competition [5] and the 2018 PhysioNet Challenge [6], which indicates the increasing interest in constructing computing methods for analyzing EEG, especially in downstream classification tasks.

In 2023, the PhysioNet Challenge [7–9] released a large corpus of data consisting of EEG, ECG (electrocardiogram), and other clinical data from comatose cardiac arrest patients from seven hospitals across the US and Europe. The EEG and other signal channels were gathered hourly, starting from ROSC and terminating mostly around the 72nd-hour mark. Its goal was to provide a computing solution for predicting the neurological outcomes of these patients using the early EEG, ECG, and other data provided and evaluating with the ground truth labels gathered through phone calls or chart review 3 to 6 months after ROSC.

EEG is very time-dependent as the signals highly fluctuate over time. Furthermore, the 2023 PhysioNet Challenge dataset contains an hour-long EEG recorded for every patient up to 72 hours after ROSC. Thus, in this thesis, we utilized this dataset in our solution that aims to maximize attention-based learning of long distance temporal dependencies among the continuous EEG recordings. Then, we used an external private dataset from the National Taiwan University Hospital (NTUH) to test the model's generalizability in a different clinical setting. In the next section, we'll first discuss all related works to this study, then show our proposed solution in more detail in the subsequent section.

doi:10.6342/NTU202403906

1.2 Related Works

In the past, researchers have used EEG to predict the neurological outcomes of coma patients through various statistical methods [10,11]. Wennervirta et al. [10] gathered EEG data from 30 coma SCA patients from the ICU of the Helsinki University Hospital and used the chi-square test to predict outcomes with clinically interpretable features such as burst-suppression ratio, response entropy, state entropy, and wavelet subband entropy as inputs. Cloostermans et al. [11], similarly, gathered EEG data from 56 coma SCA patients from the ICU of the Medisch Spectrum Twente, Enschede, the Netherlands. They built their predictive model using absent short-latency (N20) SSEP as input due to its known good feature representation of EEG [12]. Both studies obtained prediction results that indicated good discriminative abilities, differentiating the good from the bad outcomes, when evaluated with early EEG, particularly EEG from the first 24 hours after ROSC, and using automatically selected 5-minute epochs from every hour of recording.

Recently, researchers have chosen to transition from conventional statistical methods to machine learning and even deep learning methods [13–17]. A study [18] analyzed EEG data from 69 comatose SCA children, selecting the first artifact-free 5-minute epoch per hour from all available recordings. EEG recordings were first filtered with a Butterworth bandpass filter [19] in the range of 0.1 to 50 Hz, followed by a notch filter of 60 Hz to remove the power line along that frequency. Then, they selected 8 quantitative EEG (QEEG) features, including the spectral density, normalized band power along the 5 frequency bands - δ (0.5-3Hz), θ (4-7 Hz), α (6-12 Hz), β (13-30 Hz), and γ (25-50 Hz), line length, and regularity function scores. Together with the patient's age, the 8 QEEG features were normalized, and the average was obtained for each of the EEG channels. Fi-

doi:10.6342/NTU202403906

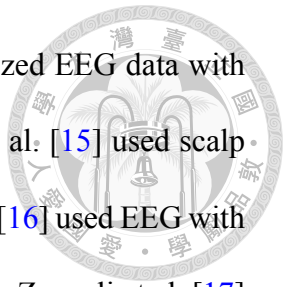
nally, they used machine learning models such as random forest, logistic regression, and support vector machine for two experimental setups, one for early EEG (0-17 hours after ROSC) and one for late EEG (18 hours onward).

Another study [13] utilized the dataset from the 2023 Physionet Challenge and used a bidirectional long short-term memory (bi-LSTM) model to learn long and short-term time dependencies. They used nine clinically interpretable features, including burst suppression ratio, Shannon entropy, δ (0.5-—4 Hz), θ (4—-7 Hz), α (8—-15 Hz), β (16—-31 Hz) band power, α/δ ratio, regularity, and spike frequency as their input to the model. They first performed bipolar referencing as it is a common method to reduce channel-wise artifacts [20, 21]. Then, they subdivided the EEG sequences into 5-minute segments and calculated the artifact scores for each 5-second segment within these 5-minute intervals, using them as weights to define signal quality. They achieved an AUROC score of 0.78 at 12 hours and 0.88 at 66 hours, demonstrating that the model's performance improves over time with clinically interpretable features as input.

The Transformer model [22], initially designed for natural language processing tasks like text translations and chatbots, has recently been applied to various time-series data. Its efficiency in handling long-distance dependencies and learning temporal patterns through parallel processing with multiple heads and positional encoding makes it well-suited for these applications. For instance, Wu et al. [23] have applied the Transformer to wind speed data and achieved promising wind speed forecasting results. Another study [24] used the Transformer for multimodal data, fusing doctors' clinical notes with structured EHR data, further indicating its adaptability to diverse datasets.

When it comes to EEG, there has also been numerous research that leveraged the

doi:10.6342/NTU202403906



Transformer for classification tasks. For instance, Du et al. [14] utilized EEG data with the Transformer to develop a model for person identification. Yan et al. [15] used scalp EEG data with the Transformer for seizure prediction tasks. Guo et al. [16] used EEG with the Transformer for emotion recognition and visualization tasks, while Zeynali et al. [17] used it for motor imagery classification.

Randomly sampling an epoch from an EEG sequence, as commonly done in previous studies [18,24], is effective. However, using full EEG sequences for training Transformer-based models, as demonstrated in several studies [14,25,26], avoids the potential waste of valuable biological data inherent in sampling only small windows, especially from long sequences of recordings.

In the 2023 George B. Moody PhysioNet Challenge, some studies [27, 28] utilized the Transformer to predict neurological outcomes, but both studies only used randomly selected 5-minute epochs. Both studies could not receive a final evaluation in the challenge due to technical problems. In our previous study [29], we also used the Transformer as our predictive model and achieved competitive results in the hidden validation and test sets in the challenge. Our previous study used features extracted from entire EEG sequences but only used the last hour of EEG recording from every patient, and both the clinical and EEG data were used as input for the model. The challenge winner [30] used a non-time-dependent model using an ensemble of machine learning models and used both EEG and ECG data to train and evaluate their models.

Some of those previous studies that used the Transformer with EEG [14–17] have used raw data as input to train their models. However, their EEG data consisted of only a few seconds to a few minutes of recordings from small tasks or events. In our study,

doi:10.6342/NTU202403906

the recordings are significantly larger since they are continuous EEG recordings recorded every hour up to 72 hours after ROSC. Therefore, it is vital to perform some feature engineering, such as what previous studies [10, 11, 13, 18] used to reduce computational complexity.

The above-mentioned previous studies have mostly used multiple types of quantitative features extracted from EEG. However, some studies [31–34] have also chosen to only use one type of feature, particularly power spectral densities (PSD), for their EEG analysis tasks and achieved promising results, indicating its excellent representation as EEG features. Alam et al. [31] used PSD features extracted from the BCI competition IV, dataset 2b, and trained an LDA classifier for motor imagery classification. Kim et al. [32] used PSD features extracted from both datasets 2a and 2b from the BCI competition and achieved good accuracies for single session, session-to-session, and the different types of 2-class motor imagery for different subjects. Wang et al. [33] extracted PSD features from 14 patients' EEGs from the Department of Neurology of Beijing Hospital for the classification of Alzheimer's disease and achieved a promising accuracy during evaluation. Finally, Dressler et al. [34] conducted a study to test patients' awareness through EEG monitoring. They evaluated the extracted PSD features from their 8-second EEG segments and used remapped prediction probability (rPK) values to compare results from different frequencies, with their best results at 35 to 127 Hz. Following these studies, we explore utilizing only PSD-based EEG features as it provides a good quantitative representation of EEG that is both clinically interpretable and provides good discriminative features for deep learning models to learn from.

doi:10.6342/NTU202403906

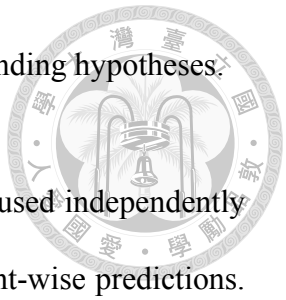
1.3 Problem Statement

In the related works, most studies were shown to only utilize 5-minute epochs to represent entire EEG sequences. However, it is intuitive that we are wasting valuable, rich biological data when we only subsample from a small sequence, especially on long continuous EEG sequences such as the one to be used in our current study. Moreover, most studies in the related works, especially those studies from the challenge [27–30], trained their predictive models using patient-wise samples, which means that each input is a recording from a single patient. When training deep models such as what we aim to use in our study, it is generally better to use more data as training input to help the model learn more patterns and generalize to new unseen data. Finally, the studies from the challenge all leveraged multimodal data to train their models and achieved very good results. However, whether we can achieve the same promising results when we utilize only EEG as the input remains unclear.

1.4 Objectives

EEG is very time-dependent as the signal varies over time for each hour of EEG recording, and the EEG from the dataset we used consisted of continuous hour-long sequences. In this thesis, we explore using an attention-based deep learning approach, particularly the Transformer [22], to predict the neurological outcomes of coma patients. The Transformer works well with time-series data due to its attention mechanism and positional encoding, which aid the model in learning long-distance dependencies across long sequences and temporal dependencies for each time position in a sequence.

doi:10.6342/NTU202403906



This thesis consists of the following objectives and our corresponding hypotheses.

1. **Recording-wise Training Method:** Each EEG sequence was used independently to train the model while the models were evaluated with patient-wise predictions. This was done mainly to increase the sample size of our model while also aiding the model in learning recording-specific patterns so it can learn to predict outcomes regardless of which hour of EEG is used during model evaluation.

2. **Capturing Long-Distance Temporal Patterns from Continuous EEG:** Each EEG sequence was initially subdivided into multiple epochs to serve as time steps, where an epoch corresponds to an event in an EEG. Often, it is a common preprocessing strategy when dealing with EEG to subsample an epoch from a sequence of EEG and only use it for analysis. In our study, we used all the hours of EEG as training samples for our model to learn important long-distance temporal patterns across each recording.

3. **Leveraging Only EEG Data to Train the Model is Enough:** Instead of using a multimodal approach as what the previous studies from the challenge [27–30] have used, we focused solely on analyzing EEG to build a model that only needs EEG as input to be able to predict neurological outcomes accurately.

We hypothesize that through these steps, we are able to maximize the learning capability of the Transformer in capturing long-distance temporal patterns among each sequence through its attention mechanism.

doi:10.6342/NTU202403906

1.5 Thesis Organization

The remainder of this thesis is organized as follows. We first introduce all important background knowledge used in our study in Chapter 2. Then, we list all materials and methods used in this thesis, including the model architecture design, signal processing steps, and other data preparation and experimental setups in Chapter 3. Following this, we show the experimental results and discussions in Chapter 4 through careful experiments and analyses. Lastly, we conclude this study in Chapter 5.

doi:10.6342/NTU202403906

Chapter 2

Background

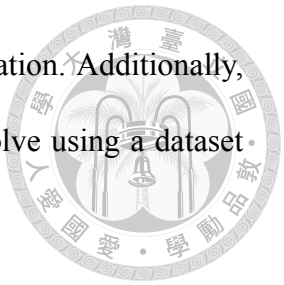
2.1 Deep Learning

A significant part of our research pipeline involves training a deep neural network. In the following sections, we will first explain the concepts to help readers understand the components used in our experiments. In order to understand deep neural networks, it is important to first discuss machine learning since deep neural networks, also known as deep learning, are a subfield of machine learning.

2.1.1 Machine Learning

Machine learning [35] can generally be divided into two main categories: supervised learning and unsupervised learning. Supervised learning involves using a labeled dataset to aid the model in learning by comparing its prediction results from the labels known as the ground truth. Unsupervised learning involves allowing the model to learn patterns across unlabelled datasets. Both types of learning are used across a large field of studies,

doi:10.6342/NTU202403906



such as in image classification, speech recognition, and machine translation. Additionally, there are also categories, such as semi-supervised learning, that involve using a dataset composed of labeled and unlabelled data.

In supervised learning, we can further divide problems into two categories: classification and regression problems. Classification involves classifying data into different classes, such as 1 or 0, for binary classification problems. Regression, on the other hand, involves generating predictions on numerical precision values, such as problems involving predictions on temperature or stock market prices. Our study uses supervised learning for a classification problem as we aim to predict patient outcomes based on early EEG, where the labels are the ground truth obtained 3 to 6 months after ROSC through phone calls or chart reviews. For supervised learning, the general formula used is $y = mx + b$, where y is the prediction, x is the input, m is the slope, and b is the intercept. The model learns by minimizing a loss function, which is generally represented by $L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)$, where N is the total number of input, y_i is the prediction for the i^{th} sample, and $\hat{y}_i$ is the ground truth for the i^{th} sample. In this formula, we are simply computing the summation of the loss (difference between prediction and label) for each sample, and by minimizing this, the model is being led to generate better predictions.

2.1.2 Deep Neural Networks

In machine learning, neural networks are computational models inspired by biological neurons in the human body. Each neuron in these artificial networks is interconnected through multiple layers, known as hidden layers, which aim to learn patterns from previous layers. Generally, the more layers a network has, the better it can learn complex patterns from the input data to make accurate predictions. A neural network that consists

doi:10.6342/NTU202403906

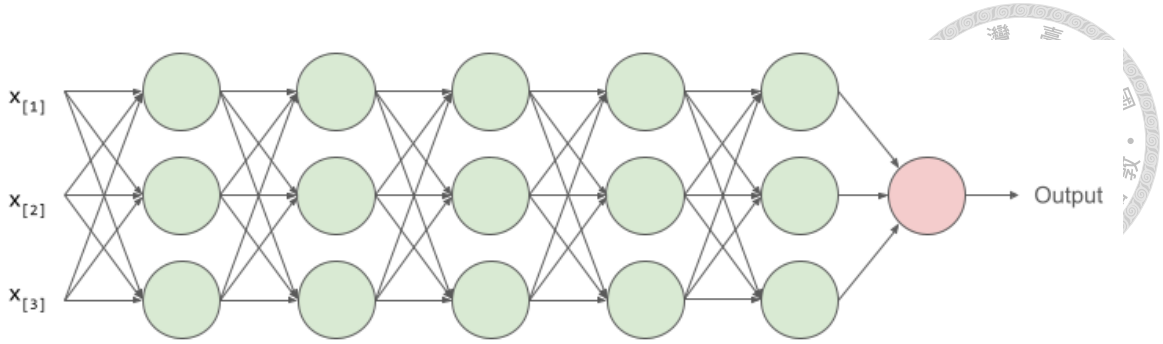


Figure 2.1: Sample diagram of a fully connected network.

of multiple hidden layers is known as a deep neural network [36, 37].

A fully connected (FC) network, also known as a dense network, is a type of deep neural network where each node in one layer is connected to every node in the next layer. This type of network architecture allows for complex relationships to be learned between input and target variables through the network layers [36]. Figure 2.1 shows a sample diagram for a simple FC network.

However, there are cases where models learn very specific patterns from the training set, leading to a situation known as overfitting. Overfitting occurs when a model, which is supposed to learn patterns that generalize well across different data types, instead learns patterns that are too specific to the training set.

The output for each hidden layer in a simple deep neural network is computed by performing a linear function then applying an activation function over the result at the k^{th} layer, denoted as $h_k = a[\beta_{k-1} + \omega_{k-1}h_{k-1}]$, where a is the activation function, β is the bias or intercept, ω is the weight vector, and h_{k-1} is the result from the previous layer or the input itself when computing the first hidden layer. In deeper neural networks, there are usually a large number of hyperparameters in the model that can be manually changed or learned during model training, and these hyperparameters play a vital role in model training to maximize the performance of the model during evaluation. The bias and the

doi:10.6342/NTU202403906

weights from the above formula are also considered hyperparameters [37].

The model learns hyperparameters that minimize the loss function through gradients that are handled by optimizers. The simplest optimizer is gradient descent, where the model initially starts with some default parameters $\phi = [\phi_0, \phi_1, \ldots, \phi_N]^T$, where ϕ is the parameter the model seeks to minimize with the loss function, that is $\hat{\phi} = argmin_{\phi}[L[\phi]]$. The first step involves computing derivatives of the loss with respect to the parameters $\frac{\delta L}{\delta \phi}$, then the parameters are updated on the way back $\phi = \phi - \alpha \frac{\delta L}{\delta \phi}$, where the scalar α determines the magnitude of the change [37].

Another popular optimizer often used in machine learning is the Adam optimizer, or 'adaptive momentum estimation.' In Adam, additional momentum and direction are incorporated into the original gradient descent equation. This change helps the model avoid the undesirable properties of gradient descent, which tend to push large adjustments to parameters associated with large gradients and small adjustments to parameters associated with small gradients. Such imbalances may cause exploding gradients and vanishing gradients, respectively [37].

Exploding gradients and vanishing gradients are phenomena that can occur during the training of deep neural networks. Exploding gradients refer to situations where the gradients become extremely large, often causing the model parameters to become unstable and leading to a failure in training. Vanishing gradients, on the other hand, occur when the gradients become extremely small, effectively vanishing and preventing the model from learning because the updates to the model parameters become negligible [36].

doi:10.6342/NTU202403906

2.1.3 Activation Function

Neural networks typically involve computing outputs using a linear function $y = mx + b$. However, deep neural networks aim to learn complex patterns, and relying solely on linear functions makes this challenging because linear functions cannot capture the intricacies of non-linear relationships in the data. For this, activation functions are used to introduce nonlinearity into the network by transforming these linear outputs. The most basic and common type of activation function is the ReLU (Rectified Linear Unit) [38], which introduces nonlinearity by retaining all positive values and transforming all negative values to zero. It is mathematically represented by $R(z) = max(0, z)$, where $R(z)$ is 0 when z is negative and z itself when it is positive. This non-linear transformation allows the neural network to learn and represent more complex patterns and functions.

2.1.4 Regularizer

Regularizers are techniques used to prevent overfitting, improve the generalization of models, and control the complexity of the model. Various methods are employed as regularizers, with some of the most common being dropout, batch normalization, and layer normalization.

One of the most common regularization techniques in deep neural networks is dropout [39], where random units (neurons) are dropped during model training. This means that a fraction of the neurons are set to zero during each forward and backward pass, effectively preventing them from participating in the computation. Dropout reduces the likelihood of overfitting by ensuring that the model does not become overly reliant on any single neuron, thereby encouraging the network to learn more robust features that generalize better

doi:10.6342/NTU202403906

to new data.

Another widely used method is batch normalization [40], which includes normalization as part of the model architecture. Batch normalization normalizes the activations of each mini-batch to have a mean of zero and a variance of one. This technique mitigates the problem of internal covariate shift, where the distribution of network activations changes during training, allowing for higher learning rates and improving the convergence speed. Batch normalization can also act as a regularizer, often reducing the need for dropout by stabilizing the learning process and improving model generalization.

Another type of regularizer is layer normalization [41], which aims to compute the mean and variance used for normalizing the samples across all the activations within a single layer rather than across a mini-batch, as with batch normalization. Layer normalization significantly improves training time and provides more stable computations across longer data sequences, making it particularly suitable for models like the Transformer that usually handle large amounts of long data sequences. Unlike batch normalization, layer normalization does not depend on the batch size and can be more effective in scenarios with varying sequence lengths or small batch sizes.

2.2 The Transformer Model

The Transformer [22] is a deep learning model that consists of an encoder and a decoder block. In general, encoder-decoder neural network architectures [42] are designed such that the encoder learns an embedding that best represents the original input sequence while the decoder transforms this embedding into a desired output sequence. Figure 2.2 illustrates a simple encoder-decoder structure, where the number of nodes in the encoder

doi:10.6342/NTU202403906

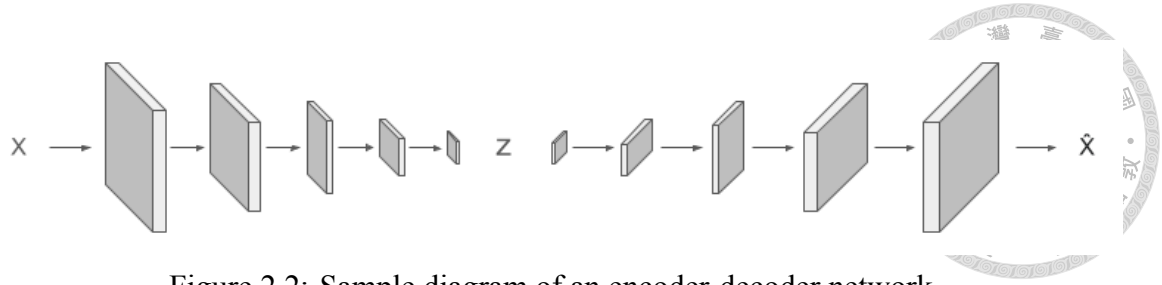


Figure 2.2: Sample diagram of an encoder-decoder network.

network reduces through each layer, and the number of nodes in the decoder network increases through each layer. Here, x represents the original input, z is the learned embedding between the encoder and decoder, and $\hat{x}$ is the reconstructed input.

However, in the Transformer, the roles of the encoder and decoder differ slightly. The encoder's objective is to learn an embedding with the same dimension as the original input, which captures essential features of the input sequence. On the other hand, the decoder uses this embedding to generate the subsequent elements in a sequence, making it particularly useful for tasks like translation and chatbots. The Transformer excels in processing long, time-series sequences due to its ability to handle parallel processing and the use of positional encodings, which help in capturing temporal patterns across time steps over long sequences. The whole model architecture of the Transformer [22] is shown in Figure 2.3.

The Transformer's encoder block consists of the multi-headed attention (MHA) and feedforward (FF) sub-blocks. First, the input sequence is divided into some number of tokens, and each token will be transformed into d_{model} sized embeddings. Then, positional encoding is injected into each token to aid the model in learning temporal patterns across the sequence. In the original implementation of the Transformer [22], the positional encoding is computed using sinusoidal functions, composed of sine and cosine functions of different frequencies, represented by $PE_{pos,2i} = sin(pos/10000^{2i/d_{model}})$ and $PE_{pos,2i+1} = cos(pos/10000^{2i/d_{model}})$, where pos is the position of the token in the se-

doi:10.6342/NTU202403906

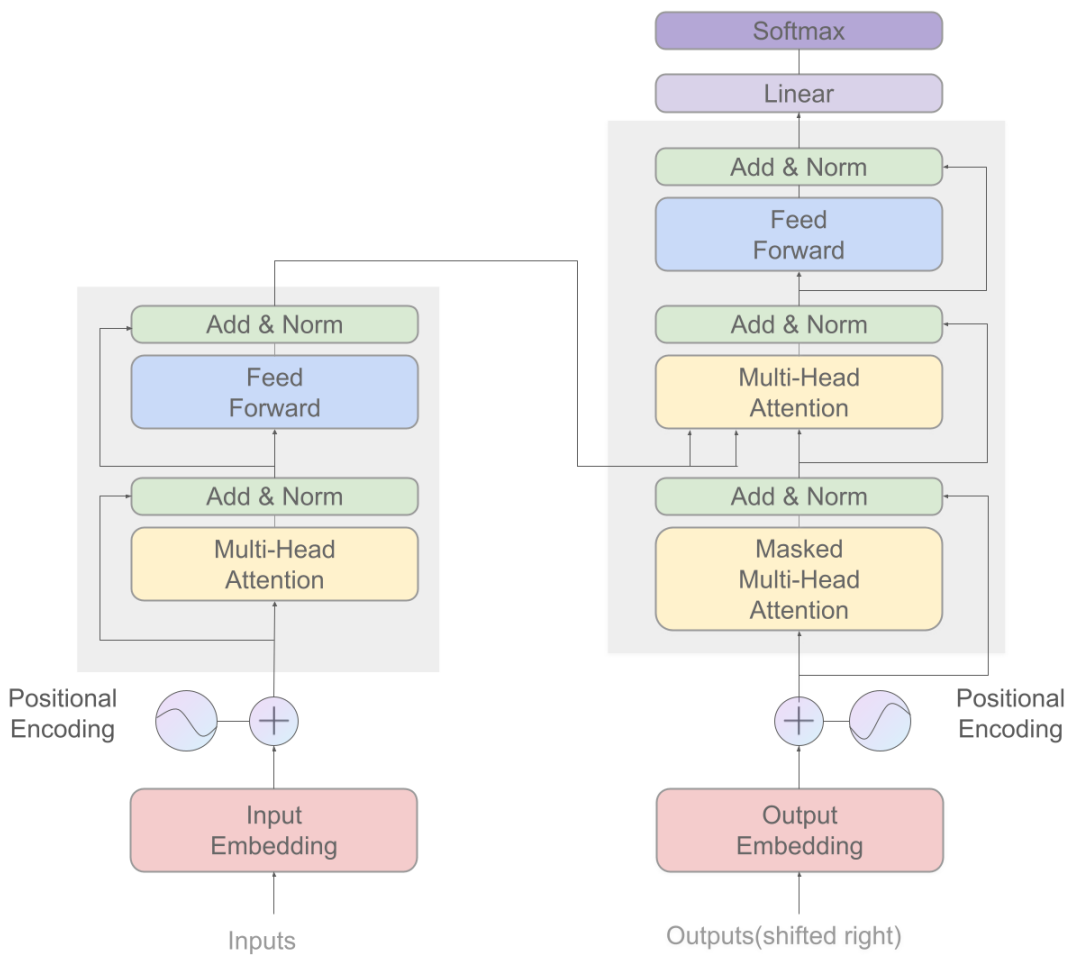


Figure 2.3: The Transformer model architecture [22].

quence and i is the dimension. Each dimension in the computed positional encoding then corresponds to a sinusoid, where the wavelengths would form geometric progressions from 2π to $10000 \cdot 2\pi$. These computed positional encodings will have the same d_{model} sized embeddings, thus allowing for convenient addition of matrices. Without positional encoding, the model will simply treat each token in the sequence equally and will not be able to identify which token comes before or after the others.

The MHA sub-block consists of a residual connection that adds the original input back to the output after MHA computation, followed by layer normalization. Residual networks have long been used in neural networks with the intention of aiding the model in avoiding losing information present from the original input by adding the input to the out-

doi:10.6342/NTU202403906

put after non-linear transformations. It was first introduced to improve image recognition tasks [43] but was later used in various models, including the Transformer. The Transformer uses layer normalization instead of batch normalization since layer normalization aims to normalize data across each layer regardless of the batch size. It also ensures stable statistics across sequences, making it well suited for the Transformer since it mostly deals with long sequences of inputs, which are processed in parallel [41].

MHA is a critical component in the Transformer, allowing the model to simultaneously attend to different parts of the input sequence. MHA is computed by first calculating three vectors: query (Q), key (K), and value (V). These vectors are computed from the input (X) through learned weight matrices. Specifically, the computation is as follows: $Q = XW^Q, K = XW^K, V = XW^V$, where W^Q, W^K, W^V are learned weight matrices. The scaled dot-product attention, as shown in Figure 2.4, is then computed as $\textit{Attention}(Q, K, V) = \textit{Softmax}(QK^T/\sqrt{d_k})V$, where d_k is the dimension of the key vectors and serves to normalize the dot product of Q and K.

In self-attention, a crucial mechanism in the Transformer, the queries, keys, and values all come from the same source sequence. This allows the model to weigh the importance of each token in the sequence relative to every other token, enabling it to attend to different parts of a single input sequence to compute its representation. MHA extends this by projecting the queries, keys, and values into multiple subspaces (or heads) and performing the attention operation in parallel:: $head_i = \textit{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. Finally, the outputs of these parallel attention heads are concatenated and linearly transformed to produce the final output of the MHA block: $\textit{MHA}(Q, K, V) = \textit{Concat}(head_1, \ldots, head_h)W^O$, where W^O is another learned weight matrix. The diagram for computing multi-head attention is shown in Figure 2.5.

doi:10.6342/NTU202403906

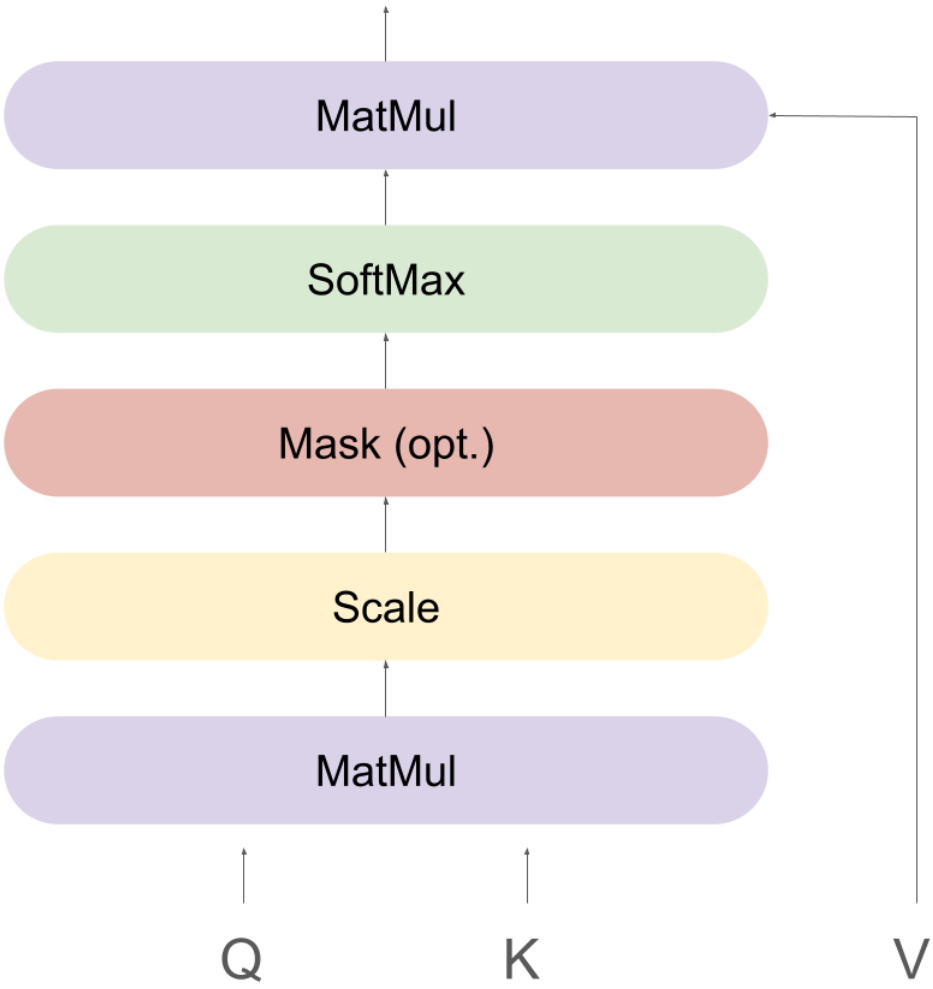


Figure 2.4: Scaled dot-product attention with optional masks to selectively ignore specific positions in the input sequence [22].

The FF sub-block in the original implementation of the Transformer [22] consists mainly of an FC network. In the Transformer, the FF sub-block consists of two linear transformations, followed by a ReLU activation in between as represented by the equation: $FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$. Here, x is the input, W_1 and W_2 are weight matrices, and b_1 and b_2 are bias vectors. The ReLU activation function ensures non-linearity by outputting zero for negative values and the input itself for positive values. These mechanisms in the Transformer's encoder block allow the model to capture different aspects of the input sequence by attending to various positions in multiple ways, enriching the input representation.

doi:10.6342/NTU202403906

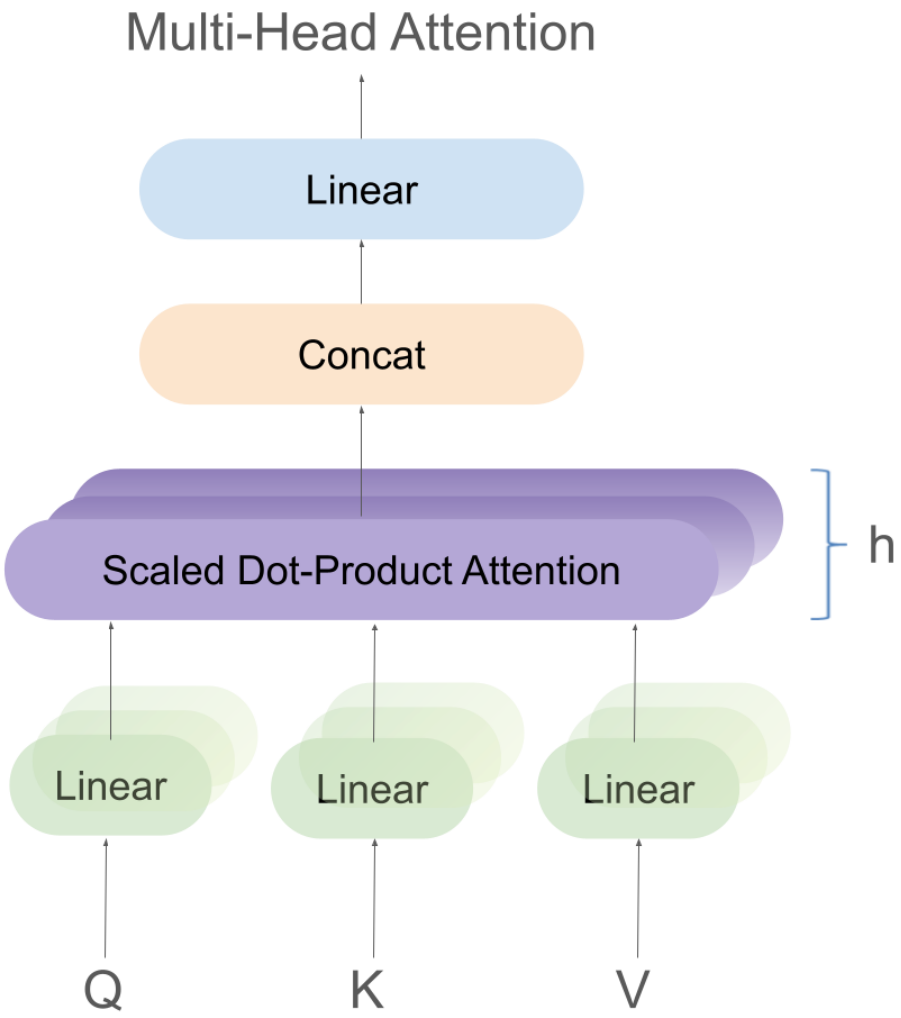
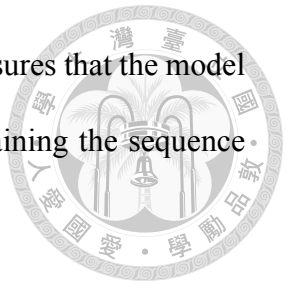


Figure 2.5: Multi-head attention with several 'h' attention layers running in parallel [22].

The decoder block of the Transformer introduces an additional sub-block compared to the encoder block, specifically a third MHA sub-block that attends to the output from the encoder block. This is in addition to the two sub-blocks present in each encoder block: the MHA sub-block and the FF sub-block. The most notable difference in the decoder block is its use of masking in the new self-attention sub-block. This masking prevents the decoder from attending to subsequent positions in the sequence. The intuition behind this is to ensure that the model does not "peek ahead" to future tokens when predicting the next token in a sequence. This is crucial for tasks like language generation, where the decoder must generate tokens sequentially and should only rely on the tokens that have

doi:10.6342/NTU202403906

already been generated. Thus, the masked self-attention sub-block ensures that the model can only consider the tokens that precede it at each position, maintaining the sequence generation's causal nature.



2.3 Evaluation Metrics

The metrics to be used throughout this study for evaluating the proposed model are as follows: accuracy, the area under the receiver operating curve (AUROC), the area under the precision-recall curve (AUPRC), and the F1 measure. These metrics have been chosen to evaluate the results of a binary classification problem, where classes are either positive, class 1, or negative, class 0. The results can be interpreted with true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), where TP are the positive samples that have been correctly predicted, FP is the positive samples that have been incorrectly predicted, TN are the negative samples that have been correctly predicted, and FN is the negative samples that have been incorrectly predicted. In this section, we will give a general overview of each metric and how it is interpreted.

Accuracy is one the most straightforward ways to evaluate a machine learning model, where $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$. This metric aims to calculate the number of correct predictions over the total number of samples. However, this metric doesn't take into consideration class imbalances. For instance, if the number of negative samples in the study is too small, then it will put more weight on the positive samples.

AUROC [44] is the resulting area below the curve when the true positive rate (TPR) is plotted against the false positive rate (FPR), where TPR is calculated as $TPR = \frac{TP}{(TP+FN)}$ and FPR is calculated as $FPR = \frac{FP}{(FP+TN)}$. This metric evaluates the degree of separability

doi:10.6342/NTU202403906

among the two classes in a binary problem. The baseline for evaluating AUROC is at 0.5, where an AUROC of 0.5 is predicted by chance, and an ideal AUROC would be valued as close as possible to 1.0.

AUPRC [45] is the resulting area below the curve when the precision is plotted against the recall. Precision, which is the ratio of correctly predicted samples from the positive classes, is computed as $\frac{TP}{(TP+FP)}$. Recall, which measures a model's capability of predicting positive classes, is computed as $\frac{TP}{(TP+FN)}$. Since AUPRC is the resulting area below the curve resulting from these two metrics, it is generally not dependent on the balance between the two classes, unlike the other metrics. However, when dealing with heavily class-imbalanced datasets, it is important to find the optimized baseline of the AUPRC to understand the true performance of the model. The equation for computing the optimized baseline is by computing the ratio of positives over the entire dataset, $\frac{TP+FN}{TP+TN}$. AUPRC values above the baseline are considered good, and the best AUPRC would be a value as close as possible to 1.0.

F1 measure is another metric used to measure the harmonic mean between precision and recall and is computed with $\frac{TP}{TP+\frac{1}{2}(FP+FN)}$. In highly class-imbalanced datasets, the threshold for computing predictions is often adjusted to maximize F1 measure performance [46]. This means that instead of the default prediction threshold, where prediction probabilities above 0.5 are classified as 1 and prediction probabilities below 0.5 are classified as 0, this decision threshold is adjusted to obtain the best F1 measure during model training. This optimized threshold is then used during model evaluation.

doi:10.6342/NTU202403906

2.4 Model Selection

In machine learning, the optimal model is typically chosen by maximizing its performance on a validation set and conducting a final evaluation using a test set. This process initially divides the dataset into training, validation, and test sets. The model is trained on the training set, and various sets of model parameters are evaluated using the validation set to determine the best parameters. These selected parameters are then used to train the final model. Finally, the test set is employed to provide a definitive assessment of the model's generalizability.

A more structured method for model selection involves cross-validation, as shown in Figure 2.6. In cross-validation, the dataset is divided into k folds. During each iteration, the model is trained on $k-1$ folds and evaluated on the remaining fold. This process is repeated k times per epoch, as depicted with each split in the diagram, with a different fold used as the validation set in each run. The average validation loss across all folds is then used to assess the model's performance. Unlike traditional approaches, cross-validation eliminates the need for a separate validation set, treating the entire dataset both as training and validation data.

2.5 PhysioNet Challenge 2023 Dataset

For the 2023 George B. Moody PhysioNet Challenge [7, 8], the International Cardiac Arrest Research Consortium (I-CARE) [9] gathered comatose cardiac arrest patients' data from seven hospitals across the US and Europe, which is summarized in Table 2.1. Overall, the dataset consists of 1020 patients, where 60% of the dataset was used as the

doi:10.6342/NTU202403906

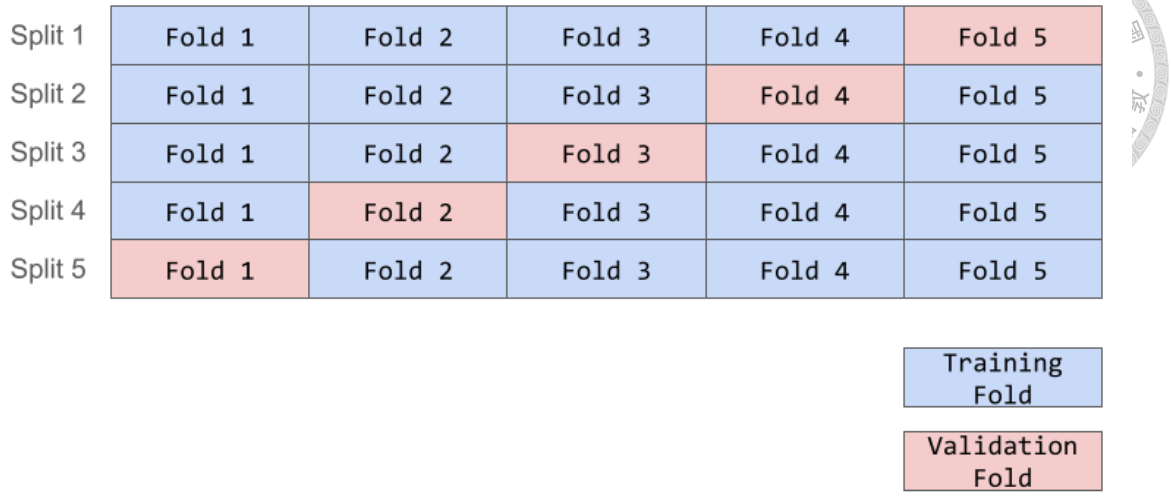


Figure 2.6: Sample diagram of cross-validation, where the training dataset is split into 5 folds for training and validation.

Table 2.1: 7 hospitals from the US and EU gathered by I-CARE.

Hospital	Country
Rijnstate Hospital	Arnhem, The Netherlands
Medisch Spectrum Twente	Enschede, The Netherlands
Erasme Hospital	Brussels, Belgium
Massachusetts General Hospital	Boston, Massachusetts, USA
Brigham and Women's Hospital	Boston, Massachusetts, USA
Beth Israel Deaconess Medical Center	Boston, Massachusetts, USA
Yale New Haven Hospital	New Haven, Connecticut, USA

training set, 10% as the hidden validation set, and 30% as the hidden test set. Patients in the dataset are either in-patient or out-patient. Only 5 hospitals were released as part of the open-source training set for 607 patients, with the remaining 2 hospitals only available as part of the hidden validation and test sets. All hospital names were de-identified to protect patient privacies, and the distribution of hospitals is shown in Figure 2.7. In this study, the entire publicly available training set was further split and used to train and evaluate the model since the hidden validation and test sets are not publicly available. Then, an external dataset from NTUH was further used to test if the model can generalize well to a new unseen dataset.

This dataset comprised clinical data and signal channels, categorized into different

doi:10.6342/NTU202403906

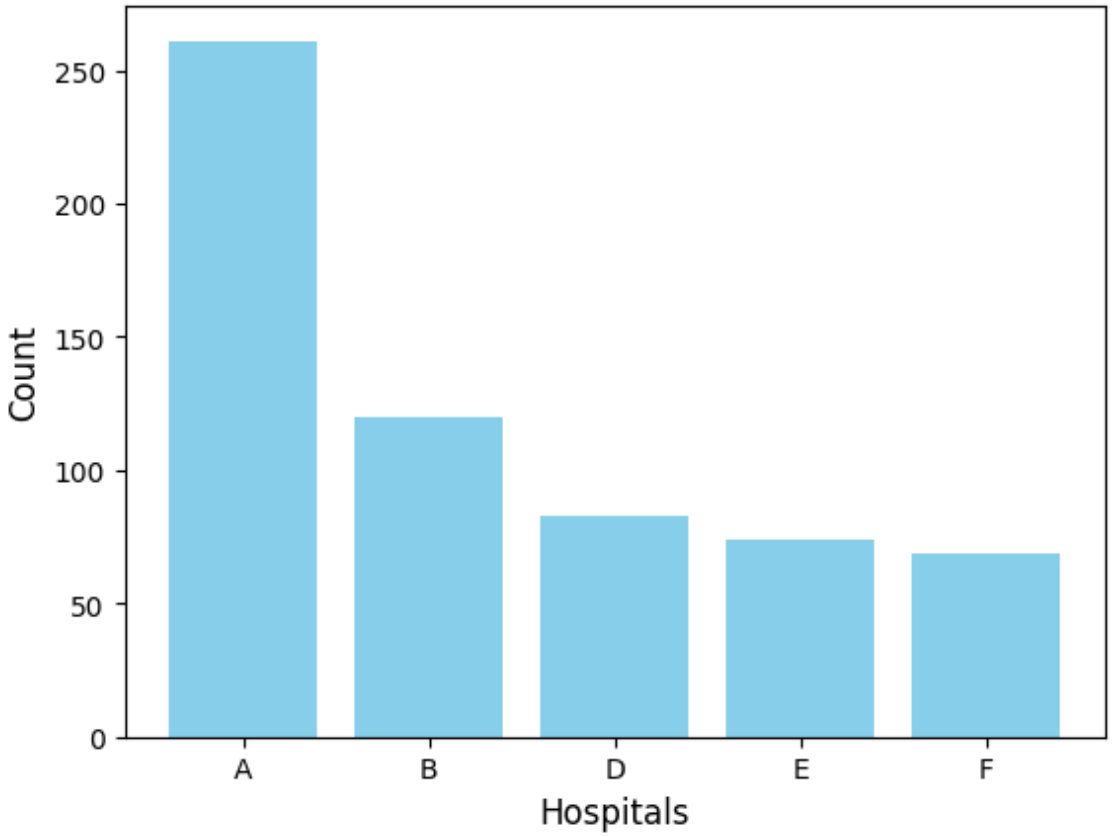


Figure 2.7: Distribution of hospitals among each patient in the PhysioNet dataset.

groups, such as EEG, ECG, REF (reference channels), AND OTHER, as shown in Table 2.2. The abbreviations for the EEG channels, which are based on the electrode's placement position in the human scalp, are summarized in Table 2.3. The clinical data in this dataset included the patient's age, sex, hospital, return of spontaneous circulation (ROSC, in minutes), out-of-hospital cardiac arrest (OHCA, true/false), shockable rhythm (true/false), targeted temperature management (TTM, in Celsius) at 33, 36, or NaN for no TTM, outcome, and Cerebral Performance Category (CPC).

CPCs were obtained 3 to 6 months from ROSC via phone interview or chart review. CPC is a widely known 5-point scale used to assess cognitive recovery. A CPC scale of 5 means death, 4 as persistent vegetative state, 3 as severe disability, 2 as moderate disability, and 1 as good recovery [47]. Outcomes were labeled as good (class 0) for CPC

doi:10.6342/NTU202403906

Table 2.2: Channels available in the PhysioNet dataset.

Channel Group	Channels
EEG	Fp1, Fp2, F7, F8, F3, F4, T3, T4, C3, C4, T5, T6, P3, P4, O1, O2, Fz, Cz, Pz, Fpz, Oz, F9
ECG	ECG, ECG1, ECG2, ECGL, ECGR
REF	RAT1, RAT2, REF, C2, A1, A2, BIP1, BIP2, BIP3, BIP4, Cb2, M1, M2, In1-Ref2, In1-Ref3
OTHER	SpO2, EMG1, EMG2, EMG3, LAT1, LAT2, LOC, ROC, LEG1, LEG2

Table 2.3: EEG Channels Abbreviations.

Channel Label	Description
Fp1	Frontal Pole 1
Fp2	Frontal Pole 2
F7	Frontal Lobe 7
F8	Frontal Lobe 8
F3	Frontal Lobe 3
F4	Frontal Lobe 4
T3	Temporal Lobe 3
T4	Temporal Lobe 4
C3	Central 3
C4	Central 4
T5	Temporal Lobe 5
T6	Temporal Lobe 6
P3	Parietal Lobe 3
P4	Parietal Lobe 4
O1	Occipital Lobe 1
O2	Occipital Lobe 2
Fz	Midline Frontal
Cz	Midline Central
Pz	Midline Parietal
Fpz	Midline Frontal Pole
Oz	Midline Occipital
F9	Frontal Lobe 9

doi:10.6342/NTU202403906

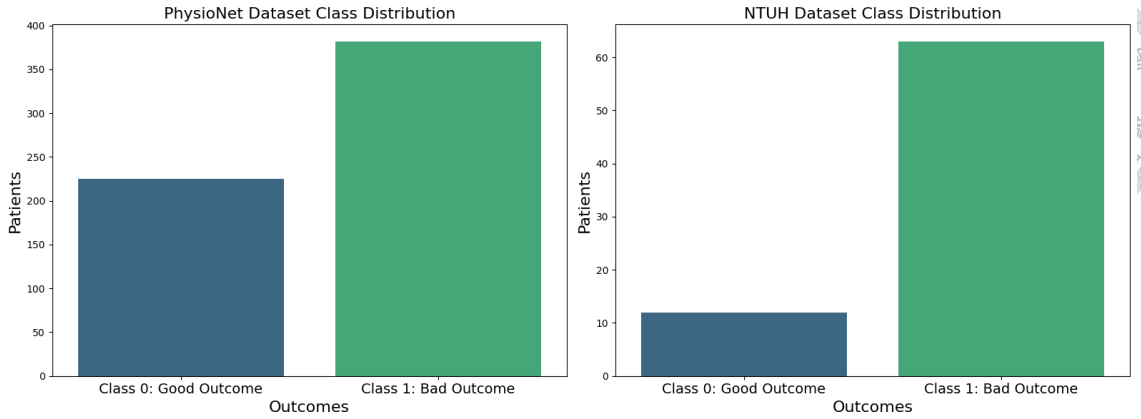


Figure 2.8: Patient outcome class distributions of the PhysioNet dataset (left) and NTUH dataset (right). Class 0 represents good outcomes, while class 1 denotes bad outcomes.

values of 1 and 2 and bad (class 1) for CPC values of 3, 4, and 5. These outcomes and CPC serve as the ground truth labels for the dataset. Our study only focuses on training our model with the outcomes as the primary label.

The outcomes class distribution per patient is shown in Fig. 2.8 and we compare it side-by-side with the class distribution from the NTUH dataset, which will be discussed in more detail in the next section. Here, it is evident that there are more patients with bad outcomes (class 1) compared to good outcomes (class 0) in both datasets. Each patient had their EEG and other signal channels recorded hourly after ROSC. Some patients recorded their continuous EEG immediately, while others started later or had to stop at certain hours due to external factors and patient conditions. Fig. 2.9 shows the distribution of hours when each hourly EEG recording was taken for all patients in the training set. Furthermore, each hourly EEG recording had differing lengths as some EEG started later for certain hours or had to terminate earlier before the hour ended.

doi:10.6342/NTU202403906

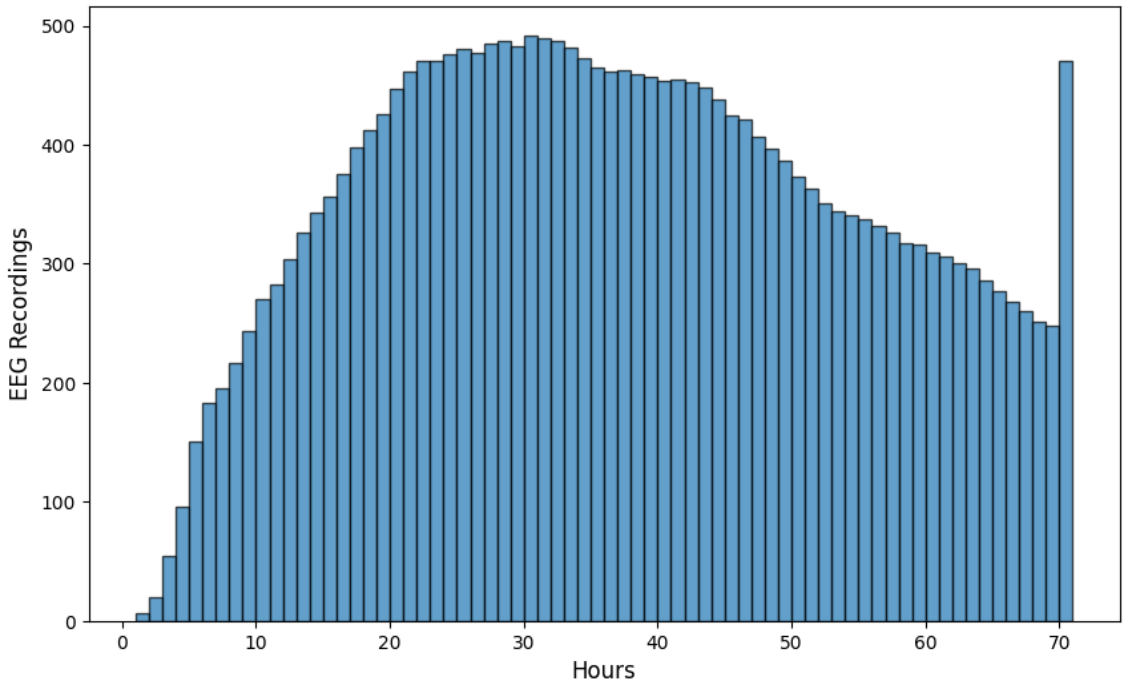


Figure 2.9: EEG recordings' hourly distribution from the PhysioNet dataset.

2.6 NTUH Dataset

The NTUH dataset, which was used in a former study [48], is a private dataset that was collected from the National Taiwan University Hospital. This dataset will be used as an external test set to evaluate the generalizability of our model. It consists of 75 comatose patients who had been resuscitated following cardiac arrest and had been comatose between 2013 and 2017 in the ICU of NTUH. Twelve patients were defined as good outcomes, with a CPC scale of 1 and 2, and sixty-three patients were defined as bad outcomes, with a CPC scale of 3, 4, and 5. This class distribution, as shown in Figure 2.8, shows an ideal clinical scenario where the bad-outcome patients will always outnumber the good-outcome patients because most coma patients eventually end up with bad outcomes in real-life clinical scenarios.

Contrary to how the dataset from the PhysioNet challenge was collected, the EEG

doi:10.6342/NTU202403906

recordings here were collected in a standard EEG room due to facility limitations. EEG data and various clinical data such as demographics and protocols like TTM were collected from each patient. The dataset contains the following EEG channels: F4-A2, C4-A2, P4-A2, O2-A2, F3-A1, C3-A1, P3-A1, O1-A1, F8-A2, T4-A2, T6-A2, F7-A1, T3-A1, T5-A1, X4-X3, Fp1-A1, Fp2-A2, Fz-A1, Pz-A1, Cz-A1. Notably, the channels all reference the A1 and A2 REF electrodes. The EEG data were all collected with a sampling rate of 200 Hz and during the 3rd and 7th day after ROSC.

doi:10.6342/NTU202403906

Chapter 3

Methods

3.1 Study Design

In this study, we used all EEG recordings from the first 80% of the patients in the publicly available PhysioNet dataset, 485 patients, to train the model. The remaining recordings from the last 20% patients, 122 patients, were used as the holdout test set to evaluate the chosen model from cross-validation during model training. The resulting class distribution of the training set is shown in Figure 3.1, with a ratio of 39.38% for class 0 (good outcomes) and 60.62% for class 1 (bad outcomes). The ratio of this training set split resembles the original class distribution of the full public dataset at 37.07% to 62.93%.

This study focuses on only EEG, so all the other channels and clinical data available in the public dataset were not used. Overall, a total of 19 EEG channels were carefully selected to match both the PhysioNet datasets and the NTUH dataset since they are the channels present in both datasets. The 19 channels are: F4, C4, P4, O2, F3, C3, P3, O1, F8, T4, T6, F7, T3, T5, Fp1, Fp2, Fz, Pz, and Cz. Following the superior predictive

doi:10.6342/NTU202403906

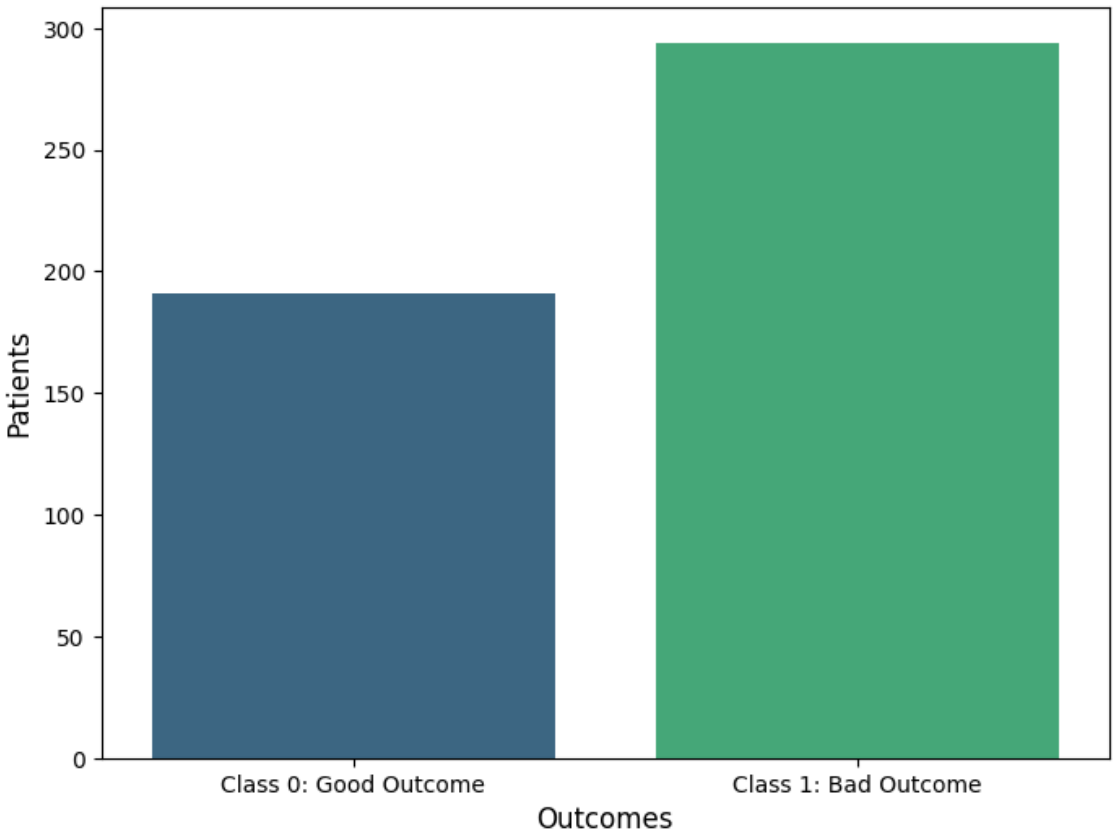


Figure 3.1: Class distribution of outcome labels in the PhysioNet training set (80% split).

capabilities of early EEG as shown in previous studies [10, 11, 49–51], we chose to use only the first 72 hours of EEG recordings. This conveniently coincides with the study's goal of aiding clinicians with patient prognosis at the 72-hour mark after ROSC [4].

We used entire hour-long EEG recordings, split into multiple time steps, to train and evaluate our model. However, the model we used, the Transformer, is a computationally expensive time-series model. Thus, to make use of the entire hour-long recordings, we first extracted clinically interpretable QEEG features, particularly power spectral densities (PSD) among frequency band–δ (0.1 to 4.0 Hz), θ (4.0 to 8.0 Hz), α (8.0 to 12.0 Hz), and β (12.0 to 30.0 Hz) from each channel, since previous studies [31–34] have utilized only PSD features to represent their EEG sequences and have achieved promising results. We extracted these QEEG features at every step of each EEG sequence. For feature selection, we opted to only use PSD features to maintain simplicity and consistency

doi:10.6342/NTU202403906

in our analysis. This is to avoid the additive noise and complexity that might result from combining different types of features as what previous studies did [10, 13, 18].

Our study mainly differs from previous studies, particularly from the challenge [27–30], through the following changes. First, the previous studies used multimodal data to train their models, one [29] using both EEG and clinical data, and another [30] using both EEG and ECG to train their models. In this study, we focused on training our proposed model solely with EEG.

Second, the previous studies all focused on patient-wise training for their models, using only 1 data per patient. In this study, we used each EEG recording as a training sample, regardless of the patient from which it came. However, the model is still evaluated using patient-wise predictions. This is done by aggregating the predictions from each recording of every patient through global average pooling (GAP). This approach of using recording-wise samples allows us to increase our sample size, enhancing performance. The Transformer benefits significantly from larger datasets, as it can learn better attention weights with the addition of more data. Moreover, it also helps the model learn recording-wise patterns, allowing it to generalize unseen data better, regardless of recording time. Figure 3.2 shows our proposed method of training our model recording-wise (left side) rather than the traditional way of patient-wise (right side) model training.

Another advantage of using recording-wise training was that we were able to balance out the class distribution of the dataset used to train our model. Figure 3.3 shows the class distribution of the outcome labels from each EEG recording. Here, we can observe a more balanced distribution of classes that may allow the model to learn better patterns from each class more equally.

doi:10.6342/NTU202403906

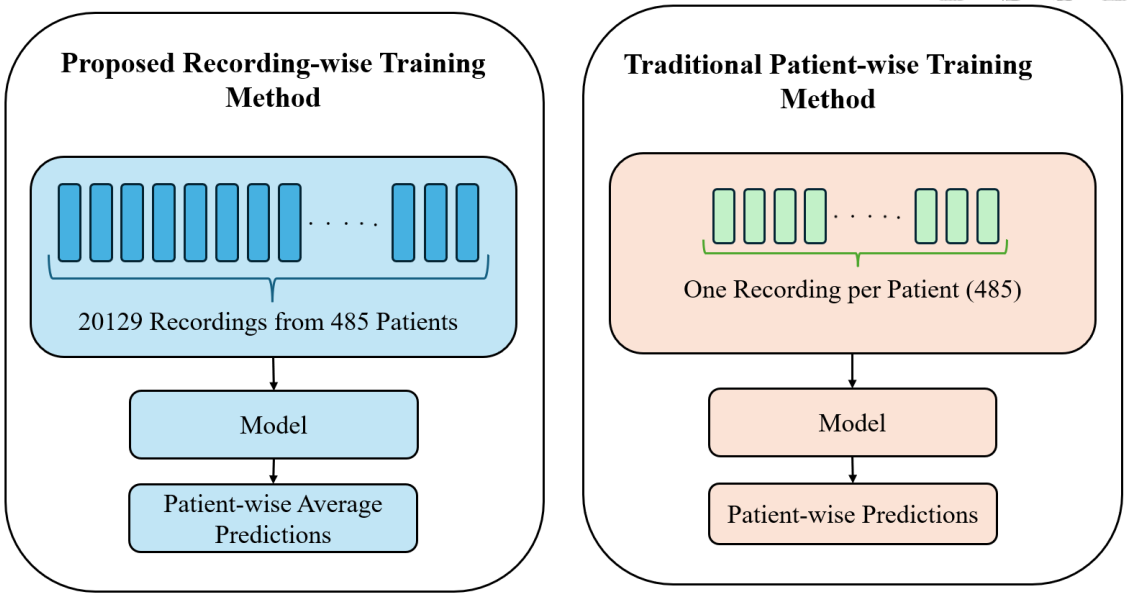


Figure 3.2: Our proposed recording-wise training vs. traditional patient-wise training method.

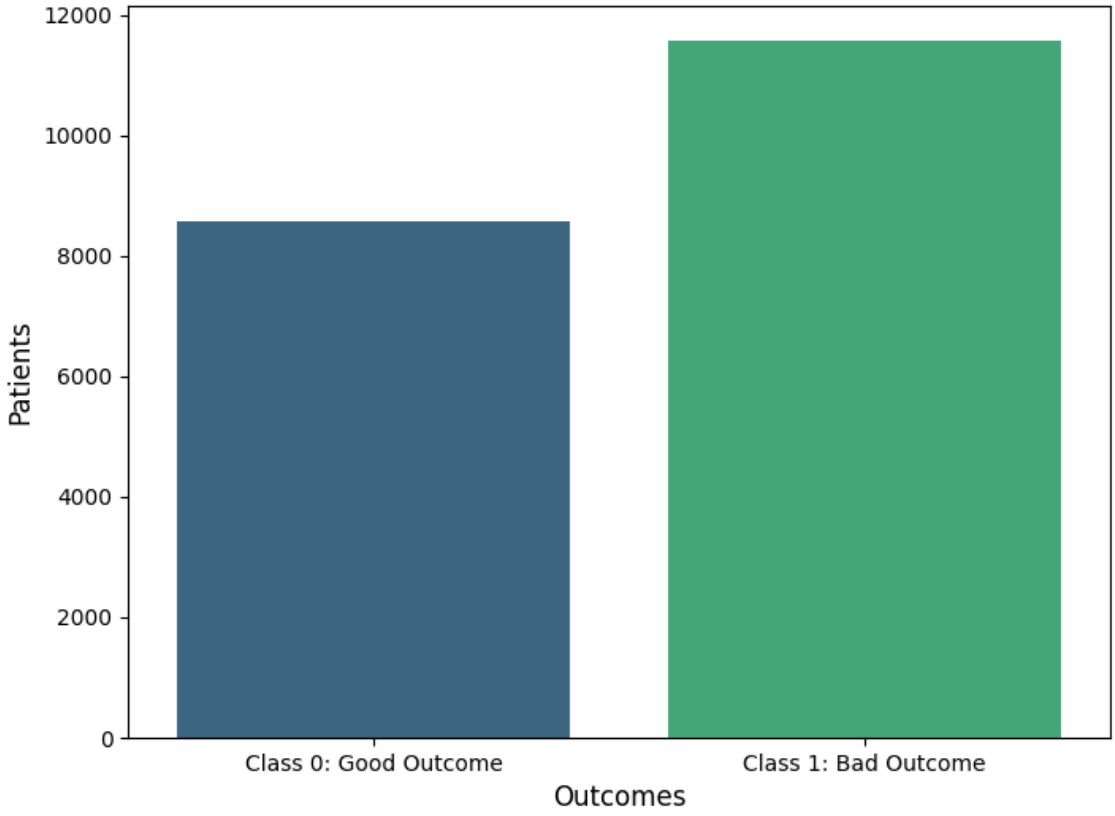


Figure 3.3: Class distribution of individual EEG recordings in the PhysioNet training set

Third, instead of subsampling a random 5-minute epoch from each continuous hour of EEG as what the previous studies [27, 28] did, we used full sequences of hour-long EEG recordings, segmented into 5-minute epochs. We hypothesize that this approach would

doi:10.6342/NTU202403906

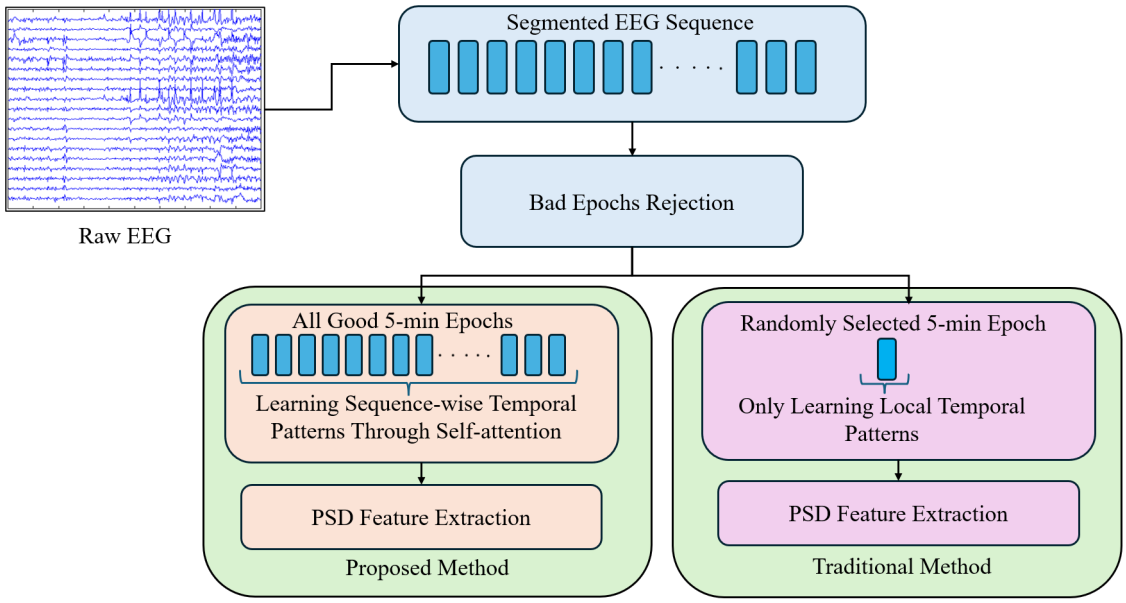


Figure 3.4: Maximizing attention-wise learning by using entire sequences (lower left), compared to traditional random epoch selection (lower right).

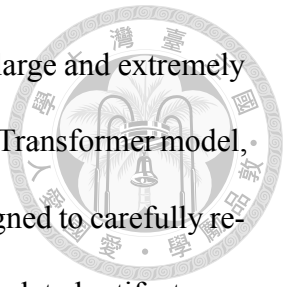
allow our model to learn better attention weights and make more accurate predictions by capturing temporal patterns across time steps from the entire continuous EEG recordings rather than local patterns within a subsampled epoch. Figure 3.4 shows a side-by-side comparison of our proposed method of using entire sequences with the traditional method of processing EEG using a subsampled epoch from the sequence.

3.2 Signal Processing and Feature Extraction

Following the study design, we used a total of 20129 EEG recordings from the first 485 patients to train our proposed model. Each EEG recording, captured within 72 hours after ROSC, was collected and treated as independent data. We utilized the 19 EEG channels selected from the study design. The recordings were divided into multiple 5-minute epochs, which served as the time steps or tokens for our Transformer-based model.

We used the MNE library [52] to perform signal processing on our EEG sequences.

doi:10.6342/NTU202403906



Signal processing [53] is crucial because continuous EEG signals are large and extremely noisy. However, since this study focused on experiments involving the Transformer model, we performed only basic signal-processing steps. These steps are designed to carefully remove unwanted frequencies and general channel-wise and recording-related artifacts.

Bad 5-minute epochs were automatically dropped by the MNE library. Then, band-pass filtering of [0.1, 30] Hz was used to filter out the unwanted frequencies. 30 Hz was used as the low-pass filter since this study used EEG signals from coma patients, where frequencies above 30 Hz, γ frequency, are generally unwanted frequencies. These signals represent high cognitive processes that are absent in coma patients.

Then, the EEG recordings were resampled from the original sampling rate of 500 Hz to 128 Hz. This step is intended to reduce computational load and storage requirements while retaining the essential information needed for analysis. Resampling simplifies the data without significantly compromising signal quality, making it more manageable for processing and model training. According to the sampling theorem, "A signal can be exactly reproduced if it is sampled at a frequency greater than twice the maximum frequency present in the signal [54]." Given that the maximum frequency in the EEG signals is only 30 Hz, a resampling rate above 60 Hz (2 * 30 Hz) can accurately reproduce the original signal. Thus, resampling to 128 Hz ensures that we do not lose important data crucial for the model to learn from.

Afterward, the EEG signals were normalized to values from –1 to +1. This step is to ensure the normalized signals are all on a common scale. Additionally, because it centers the signal around zero, it helps obtain more accurate spectral estimates since the next step after this is extracting spectral features. Figure 3.5 summarizes the entire signal-

doi:10.6342/NTU202403906

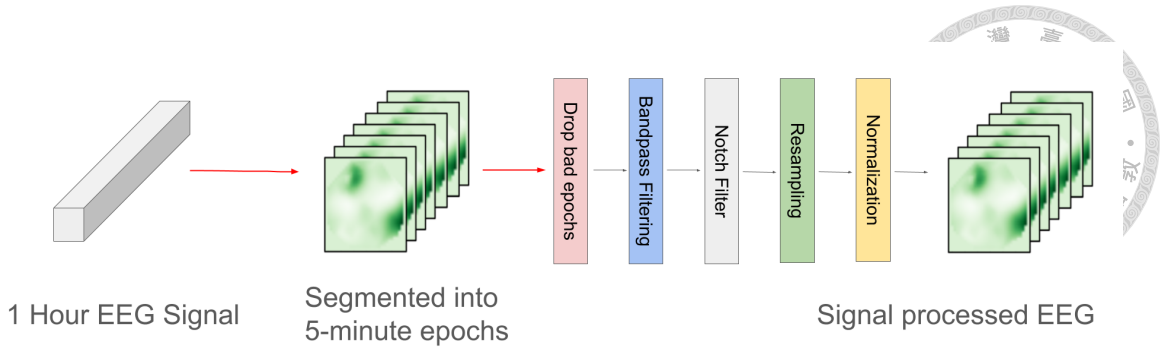


Figure 3.5: Signal processing pipeline.

processing pipeline.

The next step after signal processing is feature extraction, which is summarized in Figure 3.6. The PSD was computed from every channel in each 5-minute epoch within each EEG sequence and was further split into frequency bands–δ (0.1 to 4.0 Hz), θ (4.0 to 8.0 Hz), α (8.0 to 12.0 Hz), and β (12.0 to 30.0 Hz). The process involves transforming the time-domain samples first into the frequency-domain by using fast Fourier transform with Welch's method [55]. The resulting frequency-domain samples have the same length as the original time-domain samples. The average power for each frequency band per channel was then calculated to serve as the QEEG features for the 5-minute epoch. Finally, the mean PSD features for each frequency band per channel were concatenated into a single feature vector, totaling 76 quantitative EEG features (19 EEG Channels $\times$ 4 Frequency Bands $\times$ 1 mean PSD $=$ 76 features) which represents the EEG feature of the corresponding 5-minute epoch within each EEG sequence. These features were subsequently used as the input for our model.

The same signal-processing steps were used for both training and test sets, including the PhysioNet and the NTUH datasets. However, it must be noted that the NTUH dataset utilized EEG channels referenced to REF electrodes, and the specific voltages used for these REF electrodes are not available. Therefore, we performed average re-

doi:10.6342/NTU202403906

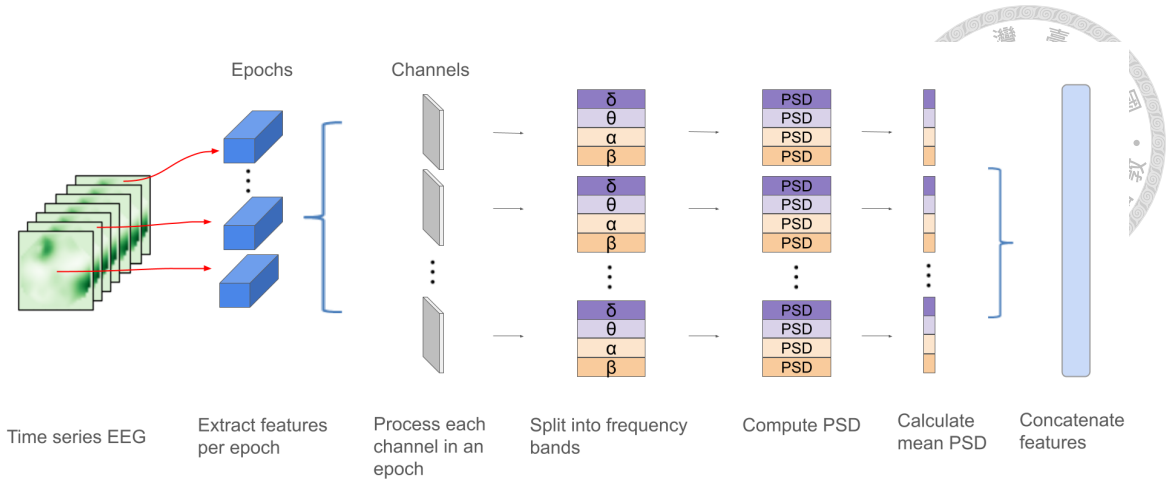


Figure 3.6: Feature extraction pipeline.

referencing [56] to reduce the contribution of common noise or artifacts that affect all channels similarly. Average re-referencing is a method wherein the average voltage across each channel is subtracted from each channel to approximate the original voltages for each channel. The first step in average re-referencing is to first calculate the average voltage across each channel by $V_{avg} = \frac{1}{N-1}\Sigma_{i=1}^{N} V_{original}(i)$, where V_{avg} is the average voltage, N is the total number of EEG channels, and $V_{original}(i)$ is the original voltage at channel i. Then, average re-referencing per channel can be computed as $V_{new}(i) = V_{original}(i) - V_{avg}$, where $V_{new}(i)$ is the new voltage at channel i after re-referencing, $V_{original}(i)$ is the original voltage at channel i, and V_{avg} is the average voltage.

3.3 Preparation of EEG Time-Series Data

In cases where recordings were shorter due to pauses or discontinuations caused by clinical factors, the missing 5-minute epochs were padded with zeros. For instance, if a recording started later within an hour, such as 20 minutes into the hour, the first four 5-minute epochs were padded with zeros. Conversely, if a recording started at the beginning of an hour but ended early, zero padding was applied after the recorded data.

doi:10.6342/NTU202403906

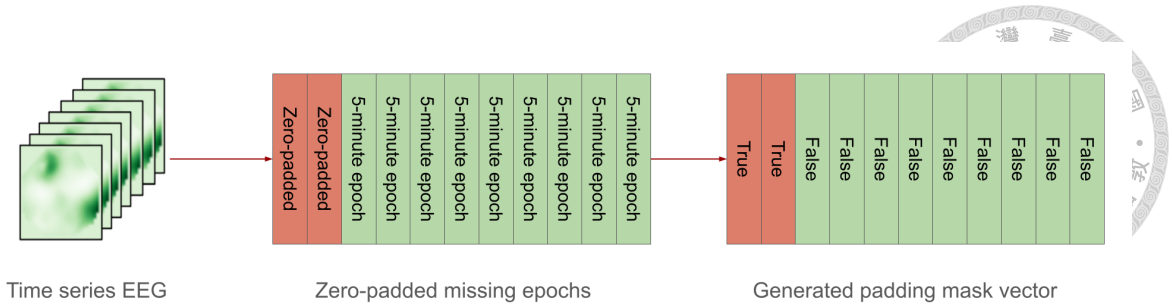


Figure 3.7: Diagram showing how zero padding was used to fill in the missing earlier epochs of an EEG recording that started 10 minutes late, along with a computed padding mask vector.

For instance, when a recording was stopped and resumed within the same hour, the two segments were simply concatenated, as they represent continuous data from the same hour. This careful preparation of time-series patterns in the EEG sequences ensures that the hourly EEG sequences maintain their realistic time steps and accurately reflect the temporal dynamics of the recorded brain activity, providing a robust foundation for training the Transformer model.

Padding masks were computed for each EEG sequence to aid the model in avoiding learning from zero-padded epochs by assigning lower attention weights to the zero-padded ones. All padded epochs will be assigned True to indicate that they were zero-padded, while the epochs that contain EEG samples will be assigned False. The resulting dimension of the padding masks is (n_recordings, n_masks), where n_recordings is the number of EEG recordings and n_masks is the same length as the number of epochs per recording since it serves as their masks. Figure 3.7 shows an example diagram for processing an EEG sequence that started recording after 10 minutes. Zero padding was applied to the first two missing epochs, and the computed padding mask vector is shown to contain True values for the zero-padded epochs.

Following these steps, the resulting dimension of our dataset is (n_recordings, n_epochs,

doi:10.6342/NTU202403906

n_features), where n_epochs is the number of 5-minute epochs within each recording and n_features is the number of features in each epoch. Prior to model training, the PSD features for all data were normalized to values from 0 to +1 through a min-max scaler. This was done to enhance convergence speed and numerical stability during model training.

The same preparation steps were used for both training and test sets, including the NTUH dataset. Both training and test sets had no missing values since bad epochs were automatically dropped. Additionally, the same scaler fitted with the training set was used to transform all test sets to prevent data leakage.

3.4 Model Training

Our proposed model, shown in Figure 3.8, consists of the Transformer's encoder block [22] to output embeddings through learned attention-weights, followed by GAP to aggregate the time step embeddings, then FC layers to aggregate the embedding vector into a final prediction output. The model is trained for multiple epochs and in batches of inputs through cross-validation.

Each EEG sequence is treated as an input sequence to the Transformer encoder, with the 5-minute epochs serving as the tokens, representing the time steps for the model. The input dimension to the model is (n_batch, n_recordings, n_epochs, n_features), where n_batch is the batch size used for training, n_recordings is the number of EEG recordings, n_epochs are the time step tokens, and n_features are the extracted EEG features used as the input.

Prior to the Transformer encoder, the EEG features are first encoded into embeddings through a linear embedding layer. Then, positional encoding, computed using sinusoidal

doi:10.6342/NTU202403906

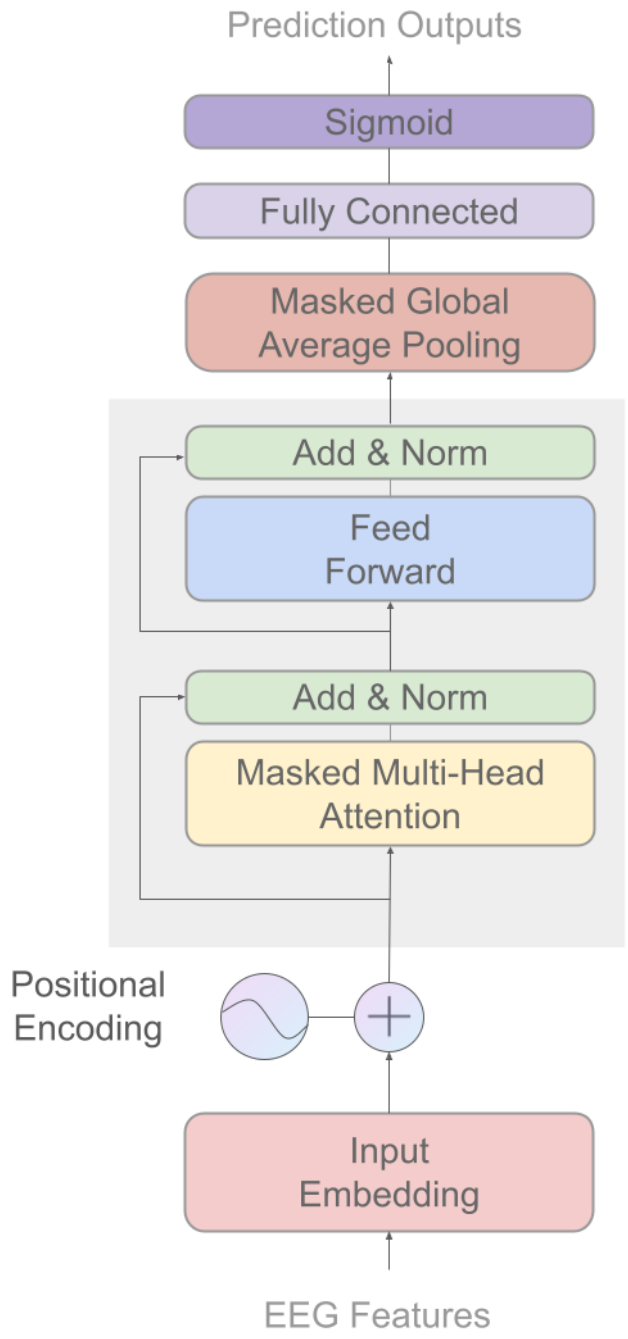


Figure 3.8: Our proposed Transformer-based [22] model architecture for downstream classification of EEG features to prediction outputs.

functions (sine and cosine functions of different frequencies) as described in the original implementation [22], is added to the input to allow the model to learn temporal dependencies within each token through their positions in the sequences.

The Transformer encoder can be divided into the MHA sub-block and the FF sub-

doi:10.6342/NTU202403906

block. To compute self-attention in the MHA block, the query, key, and value vectors all take the same embedding as their input. MHA is then computed using $\mathit{MHA}(Q, K, V) = \mathit{Concat}(head_1, \ldots, head_h)W^O$, where each head computes parallel attention along the tokens in each sequence through $\mathit{Attention}(Q, K, V) = \mathit{Softmax}(QK^T/\sqrt{d_k})V$. The final step in this sub-block consists of the residual layer where the embedding prior to the MHA block is added back to the new output embedding from MHA, followed by layer normalization. The computed padding masks from the preparation step were used here in the MHA sub-block to aid the model in assigning lower attention weights to the padded tokens. The resulting embedding of the same dimension as the original embedding prior to MHA is then passed to the FF sub-block. Please refer to 2.2 for a more detailed explanation of MHA computation.

The FF sub-block takes the output embedding from the MHA sub-block as the input. Within this sub-block, an expansion layer increases the dimensionality, followed by a ReLU activation and a dropout layer. Then, a contraction layer reduces the dimensionality back to the original size. A residual connection adds the sub-block's input embedding to the sub-block output. Finally, layer normalization is applied to the resulting embedding.

The resulting embedding from the Transformer encoder block consists of the learned embeddings of the same dimensions as the original input batch, that is (n_batch, n_recordings, n_epochs, n_embedding), where n_embedding represents the learned embedding of the same size as the original input embedding prior to MHA. Afterward, GAP is performed along the epochs or tokens for each sequence in the batch. Through the computed padding masks from the preparation steps, the padded tokens are carefully omitted during the calculation of GAP to ensure that only the embeddings from the actual EEG features are used to calculate the average embedding for each sequence. After the epochs are aggregated

doi:10.6342/NTU202403906

through GAP, the resulting dimension is (n_batch, n_recordings, n_embedding), where each recording will now only have one embedding.

Then, the resulting batch of outputs is passed to the FC block that consists of 3 contraction layers to eventually leave a single output for each EEG sequence. Finally, the sigmoid activation function transforms the outputs of each sequence into probabilities, resulting in a final output dimension of (n_batch, n_recordings, pred_prob), where pred_prob is the recording-wise prediction probability for each recording in the batch.

The final step in model training is to aggregate the recording-wise predictions, collected from the validation folds during cross-validation, into patient-wise predictions to obtain meaningful model evaluations. After every training epoch, the recording predictions from each patient are collected and averaged to obtain the aggregated patient-wise predictions. Then, the chosen metrics, AUROC, AUPRC, accuracy, and F1 measure, are used to evaluate the model through these predictions.

Since the datasets are highly imbalanced, as shown in Figure 3.1, the optimal threshold for classifying prediction probabilities is adjusted. This means that instead of using the default prediction threshold, where prediction probabilities above 0.5 are classified as an outcome of 1, and those below are classified as 0, this threshold is adjusted by maximizing the F1 measure. The optimized threshold is used to obtain the patient-wise predictions and is used to calculate both the accuracy and F1 measure. This same optimized threshold from the chosen model is used as the prediction threshold during model evaluations with the test sets.

doi:10.6342/NTU202403906

Table 3.1: Grid search hyperparameters space.

Hyperparameter	Trials
batch_size	**16**, 32, 64
num_layer	2, **3**, 4, 5, 6
dropout	0.2, 0.3, **0.4**, 0.5
embedding	32, **64**, 128
num_heads	**8**, 16, 32
learning_rate	0.001, **0.0001**, 0.00001
optimizer	**Adam**, RMSProp, AdamW, NAdam, RAdam
loss_function	**BCELoss**, BCEWithLogitsLoss
scheduler_step_size	**50**, 100

3.5 Experimental Setup

The proposed model was trained using cross-validation with k=5 across the entire training set. The final model is chosen through early stopping criteria, defined as when the patient-wise results' AUPRC from the validation results has not improved beyond the current best AUPRC for 10 consecutive epochs. AUPRC was chosen as the main metric for early stopping criteria since it is the metric among the chosen ones that is most suited to evaluate models trained with highly imbalanced datasets.

A careful search of hyperparameters was performed through grid search. Table 3.1 shows the hyperparameters grid searching range used in this study. Here, batch_size is the number of recordings used in each training batch, num_layer is the number of stacked Transformer encoder layers, dropout is the dropout rate used throughout the model, embedding is the embedding dimension used in the linear embedding layer, and num_heads is the number of heads used in MHA computation. To avoid overfitting, we introduced a scheduler to decay the learning rate after every number of epochs, determined by the scheduler_step_size.

Our optimized model had the following hyperparameters (highlighted in Table 3.1),

doi:10.6342/NTU202403906

Table 3.2: Platform and libraries detail.

Platform or Library	Version
CPU	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
Memory	48GB
GPU	GeForce GTX 1180 with 8GB VRAM
CUDA version	11.4
mne version	1.4.2
PyTorch version	2.0.1
scikit-learn version	1.3.0
numpy version	1.22.0
joblib	1.3.1

with the batch size of 16, number of layers for the Transformer encoder at 3, the dropout rate at 0.4, 64 embedding size, 8 heads, learning rate of 0.0001, Adam optimizer, BCE (Binary Cross Entropy) as the loss function, and scheduler_step_size of 50. We trained the model for 200 epochs, where early stopping criteria were triggered at 125 epochs. Finally, Table 3.2 shows the platforms and libraries used to train the proposed model. The models were trained using PyTorch [57], and the data preparation steps prior to model training were performed with Numpy [58]. All experiments were performed with a random seed of 15 for both PyTorch and Numpy's random seeds to ensure reproducibility of the experiments for future studies.

doi:10.6342/NTU202403906

Chapter 4

Results and Discussions

4.1 Experiments Using 80% Training Set

4.1.1 Main Results

In our main experiment, we used the recordings from the first 80% patients from the publicly available PhysioNet dataset to train the proposed model. Using the experimental setup as defined in 3.5, we evaluated it with the recordings from the last 20% patients from the PhysioNet dataset and the external NTUH dataset. Table 4.1 shows the results when evaluated with both the PhysioNet test set and the NTUH dataset.

Here, we can observe very high metrics when evaluated with the PhysioNet test set, with 0.82 AUROC, 0.90 AUPRC, 0.73 accuracy, and 0.79 F1 measure. We can also observe promising results when evaluated with the NTUH dataset, with 0.65 AUROC, 0.90 AUPRC, 0.74 accuracy, and 0.84 F1 measure. The optimized threshold for the predictions was at 0.55 and was used consistently during the evaluations with both datasets. The poorer performance of the NTUH dataset, compared to the PhysioNet test set, is as ex-

doi:10.6342/NTU202403906

Table 4.1: Results from 80% PhysioNet training set.

Metric	PhysioNet Test Set	NTUH Dataset
AUROC	0.82	0.65
AUPRC	0.90	0.90
Accuracy	0.73	0.74
F1 measure	0.79	0.84

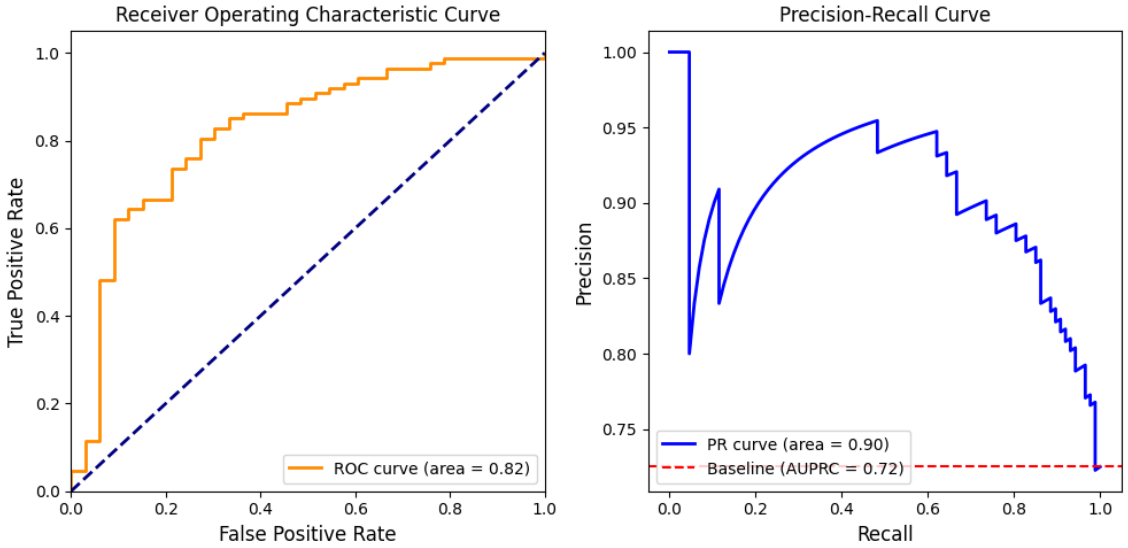


Figure 4.1: ROC (left) and PRC (right) curves when evaluated with the PhysioNet test set using the model trained with 80% training set.

pected since external test sets are meant only to evaluate a model's generalizability, and it is enough that the model can perform well.

Figure 4.1 shows the ROC (left) and PRC (right) curves when the model trained with the 80% training set was used to evaluate the PhysioNet test set, with an AUROC of 0.82 and AUPRC of 0.90. The AUPRC baseline was optimized at 0.72, which shows that the model performed very well on this test set, as the AUPRC is significantly above the baseline.

Figure 4.2 shows the ROC (left) and PRC (right) curves when the model was evaluated with the NTUH test set, with an AUROC of 0.65 and an AUPRC of 0.90. Here, the AUPRC baseline was optimized at 0.84, which shows that the model still performed well as the AUPRC is above the baseline.

doi:10.6342/NTU202403906

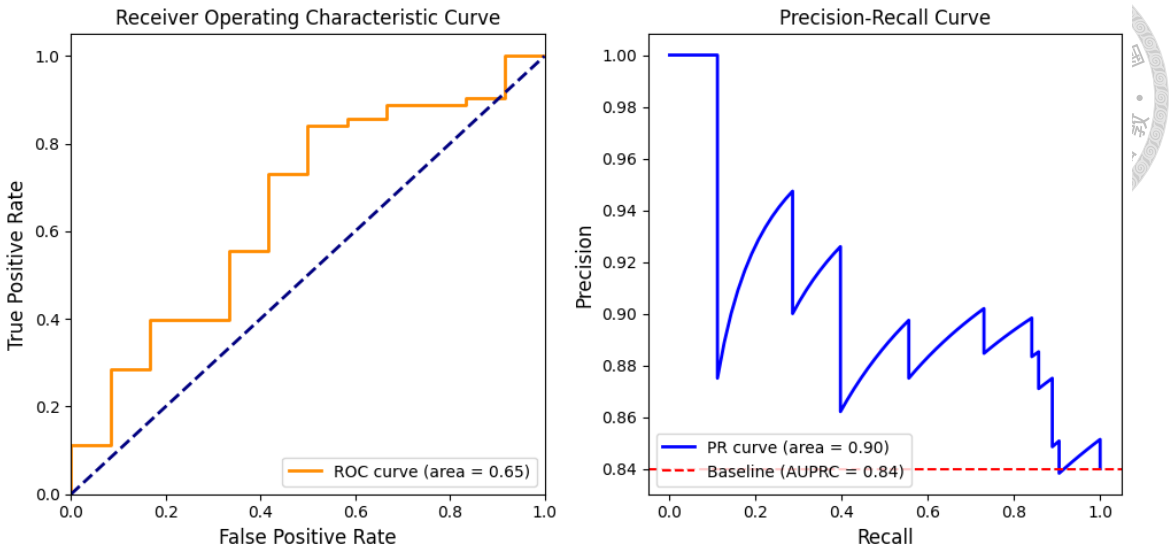


Figure 4.2: ROC (left) and PRC (right) curves when evaluated with the NTUH dataset using the model trained with 80% training set.

4.1.2 Model Evaluation at Different Hour Windows

We sought to understand how our trained model works when evaluated at certain hour windows - 12 hours, 24 hours, 48 hours, and 72 hours, to know whether the model can still perform well when evaluated only with earlier EEG recordings. This means that we tested our model with recordings from the PhysioNet test set only within 12 hours, 24 hours, 48 hours, and 72 hours for each evaluation. Figure 4.3 shows the AUROC results when evaluated with recordings at certain hour windows.

The results, when evaluated with all recordings within 72 hours, are as expected since this evaluation uses more EEG recordings. Consequently, more recording predictions are considered when aggregating the patient-wise predictions, leading to the highest AUROC performance. This also highlights the model's clinical relevance as physicians are tasked to perform patient prognoses only after 72 hours, so they can easily use this model to aid them in their task. The model's best performance at this mark reinforces its reliability and suitability for clinical use in this critical time frame.

doi:10.6342/NTU202403906

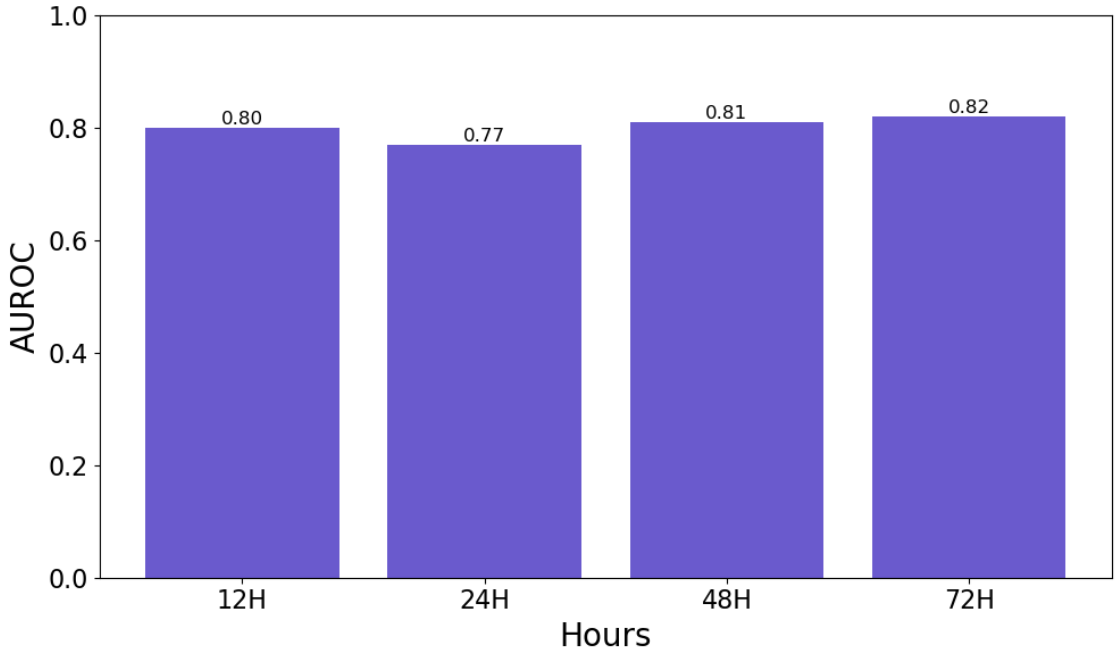


Figure 4.3: AUROC from model evaluation at each hour threshold for the PhysioNet test set.

The model generally performed well when evaluated at all hour windows, with AUROC of 0.80, 0.77, 0.81, and 0.82 when evaluated at 12 hours, 24 hours, 48 hours, and 72 hours, respectively. This experiment confirms that the model has a comparable predictive capacity at earlier and later time points, with all AUROCs remaining above 0.77.

Interestingly, the model had a slight decrease in AUROC when evaluated at 24 hours compared to when it was evaluated at 12 hours. However, this slight decrease in performance is not very significant, with only a 0.03 drop. The 0.80 AUROC, when evaluated at 12 hours, highlights the model's overall robustness even when evaluated with only early EEG recordings. This further shows that early EEG already provides clear and consistent signals, rich with information, that allow the model to learn long-distance temporal patterns through its attention mechanism. This, in turn, is useful in the clinical setting as it may help facilitate effective early risk stratification.

doi:10.6342/NTU202403906

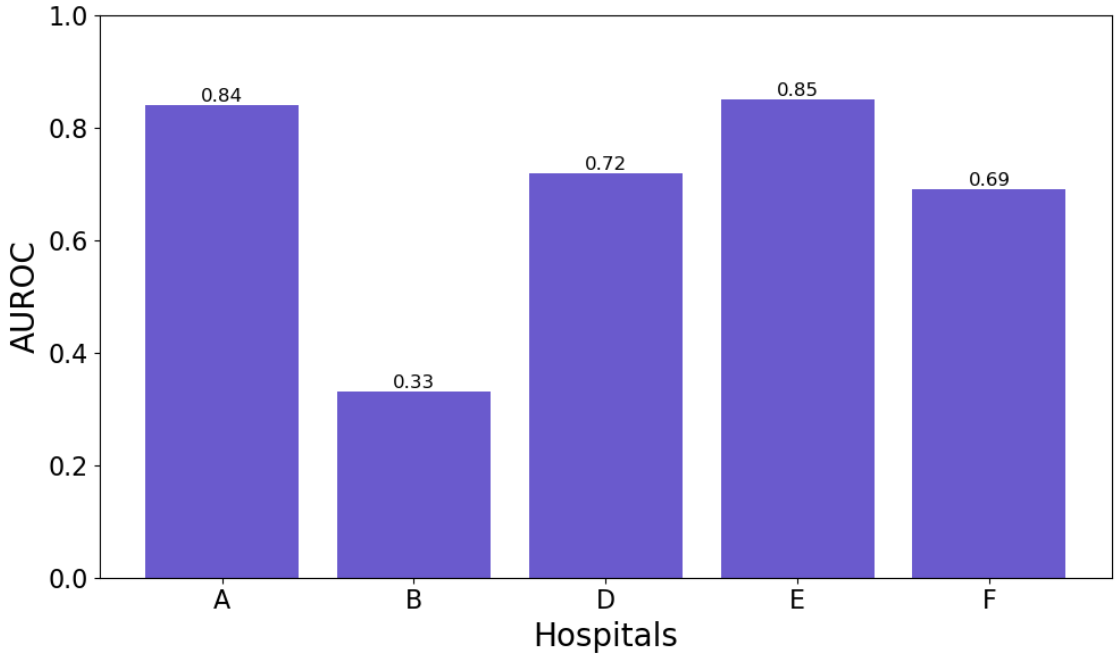


Figure 4.4: AUROC from the model evaluation on each hospital for the PhysioNet test set.

4.1.3 Model Evaluation per Hospital

We further investigated how well our model performs when evaluated with recordings from each hospital in the PhysioNet test set. Each hospital has been anonymized into alphabet letters to protect patient privacies, and two hospitals, hospitals C and G, were not made available in the publicly available PhysioNet dataset. Figure 4.4 shows the AUROC results when evaluated only with data from each hospital.

We can observe the highest AUROC of 0.85 from Hospital E, followed by 0.84 from Hospital A. As shown in Figure 4.5, the hospital with the most patients from the 80% training set is Hospital A, so it is reasonable that this hospital would have a good performance since most data comes from here. However, it is interesting that Hospital E had the best performance, even outperforming Hospital A, despite having fewer patients. The ratio of all the hospitals, as shown in Figure 4.5 across the training set, is as follows: Hospital A

doi:10.6342/NTU202403906

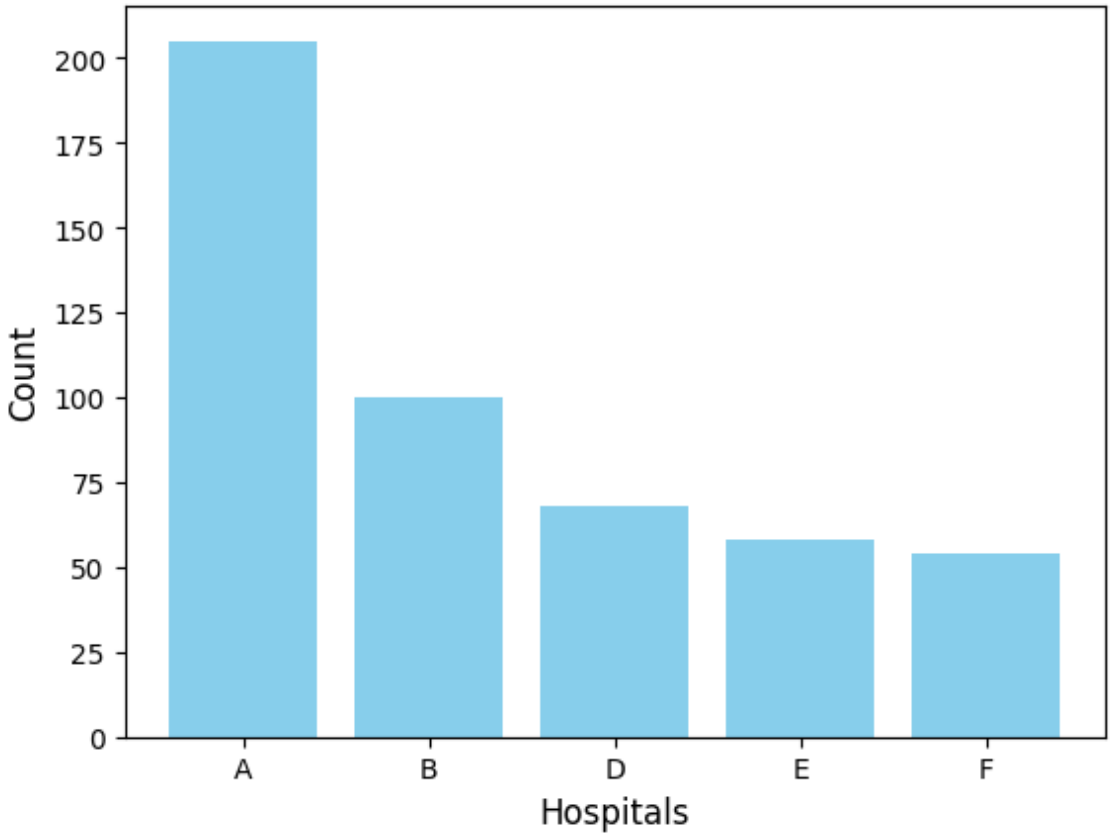


Figure 4.5: Hospital distribution per patient from the 80% PhysioNet training set.

with 42.27%, Hospital B with 20.62%, Hospital D with 14.02%, Hospital E with 11.96%, and Hospital F with 11.13%.

Among the hospitals, Hospital B had the worst performance and was the only hospital that had an AUROC below the 0.50 AUROC baseline. Future studies may focus on performing more comprehensive analyses of the data from different hospitals to understand the underlying causes of why the model performed worse and better at certain hospitals. Overall, this experiment showed that certain hospitals contributed data that benefited our proposed model more, aiding it to learn better patterns.

doi:10.6342/NTU202403906

Table 4.2: Comparison of RW and PW setups when evaluated with PhysioNet test set and NTUH dataset.

Metric	PhysioNet RW	PhysioNet PW	NTUH RW	NTUH PW
AUROC	0.82	0.58	0.65	0.61
AUPRC	0.90	0.76	0.90	0.89
Accuracy	0.73	0.71	0.74	0.7866
F1 measure	0.79	0.81	0.84	0.88

4.2 Recording-wise vs. Patient-wise Samples

In this study, we utilized each EEG recording from all patients as individual training data, regardless of which patient they come from. This diverges from previous studies, especially those from the challenge [27–30], who trained and evaluated their models using patient-wise samples. In this next experiment, we compare the results from our recording-wise (RW) model training implementation to patient-wise (PW) model training. For the PW setup, we only used the latest recording from every patient as a training sample, following the setup from our previous study [29] from the challenge. Table 4.2 presents a side-by-side comparison of results from both setups when evaluated with the PhysioNet test set and the NTUH dataset.

Comparing the two setups when evaluated with the PhysioNet test set, there was a significant drop on both AUROC and AUPRC, with 0.82 to 0.58 and 0.90 to 0.76, when evaluated with the RW and PW setups, respectively. The accuracy also experienced a small drop from 0.73 to 0.71, while the F1 measure increased slightly from 0.79 to 0.81. Similarly, when evaluated with the NTUH dataset, the AUROC and AUPRC from the RW setup were higher than the results from the PW setup, at 0.65 to 0.61 and 0.90 to 0.89, respectively. This experiment shows the value of using recording-wise predictions to train our model as the Transformer model, in particular, benefits well when trained with

doi:10.6342/NTU202403906

Table 4.3: Comparison of Full Hour and 5-minute setups when evaluated with PhysioNet Test Set and NTUH Dataset.

Metric	PhysioNet Full Hour	PhysioNet 5-minute	NTUH Full Hour	NTUH 5-minute
AUROC	0.82	0.71	0.65	0.63
AUPRC	0.90	0.83	0.90	0.91
Accuracy	0.73	0.73	0.74	0.6266
F1 measure	0.79	0.80	0.84	0.75

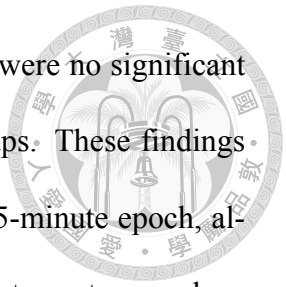
more data as it can learn better attention weights through more samples and learn better long-distance temporal dependencies across each EEG recording.

4.3 Full Hour vs. 5-minute EEG Samples

In this next experiment, we evaluated the proposed model, which was trained with full hours of EEG, with the model trained with just 5-minute randomly sampled epochs from each recording. This experiment intends to test our hypothesis that the Transformer's attention mechanism benefits well from using data from entire hourly recordings to learn long-distance relationships among each epoch in every hour. To utilize the same experimental settings for the 5-minute setup with the original setup, the randomly selected 5-minute epoch was further subdivided into thirty 10-second epochs to provide the time step tokens for the Transformer model. In the 5-minute setup, no padding mask was computed since full 5-minute epochs were simply segmented into timesteps, and this will always result in no missing epochs; hence, masking is no longer necessary. Table 4.3 shows the side-by-side results of the two setups, Full Hour for the proposed setup and 5 minutes for the setup, which uses only randomly selected 5-minute epochs from every recording.

The results demonstrate that the Full Hour setup outperforms the 5-minute setup in terms of AUROC and AUPRC when evaluated with the PhysioNet test set, achieving

doi:10.6342/NTU202403906



scores of 0.82 versus 0.71 and 0.90 versus 0.83, respectively. There were no significant differences in accuracy and F1 measure scores between the two setups. These findings suggest that using full hours of EEG data, rather than a subsampled 5-minute epoch, allows the model to maximize its attention mechanism to learn long-distance temporal relationships among time step tokens across a full hour rather than just local temporal relationships within a subsampled epoch. This is evidenced by the significant increase in overall performance when using the Full Hour setup over the PhysioNet test set.

When evaluated with the NTUH dataset, the Full Hour setup had a 0.65 AUROC, 0.90 AUPRC, 0.74 accuracy, and 0.84 F1 measure, while the 5-minute setup had a 0.63 AUROC, 0.91 AUPRC, 0.62 accuracy, and 0.75 F1 measure. These results show that the Full Hour setup generally performed better over the 5-minute setup, with the latter setup only outperforming the former over the AUPRC. However, the difference of 0.01 for the AUPRC was not very significant.

These close results, when evaluated with the NTUH dataset, maybe because the NTUH dataset only contains short recordings that do not reach an hour, and almost all the recordings had to be zero-padded for the Full Hour setup. Nevertheless, our main model, Full Hour setup, could still generalize well to this external dataset despite the difference in the number of time samples available for each recording.

These findings over the two test sets show that the Full Hour setup was superior over the 5-minute setup, as it performed significantly better with the PhysioNet test set and slightly better with the NTUH dataset.

A limitation of using full sequences of EEG recordings is that hospitals often cannot collect numerous long, hour-long sequences due to facility constraints and other envi-

doi:10.6342/NTU202403906

Table 4.4: Comparison of GAP and GMP when evaluated on the two test sets

Metric	PhysioNet GAP	PhysioNet GMP	NTUH GAP	NTUH GMP
AUROC	0.82	0.80	0.65	0.53
AUPRC	0.90	0.90	0.90	0.88
Accuracy	0.73	0.72	0.74	0.50
F1 measure	0.79	0.78	0.84	0.65

ronmental factors. However, our comparison experiment results show that our proposed model can still perform well even when evaluated with shorter recordings, showcasing its promising clinical applications.

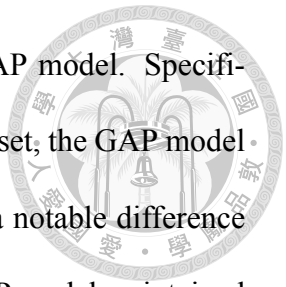
4.4 Ablation Study with Pooling Layer

One of the key layers in the model is the global average pooling (GAP) layer, which aggregates the token embeddings into a final mean embedding for each EEG sequence input. In this section, we perform an ablation study to evaluate how a different pooling strategy might affect model performance. We trained another model using global max pooling (GMP) instead of GAP to aggregate the token embeddings.

Table 4.4 presents a side-by-side comparison of the two models, evaluated using the two test sets. On the PhysioNet test set, the GAP model generally outperformed the GMP model, achieving an AUROC of 0.82 compared to 0.80, an accuracy of 0.73 compared to 0.72, and an F1 score of 0.79 compared to 0.781, while the AUPRC remained the same at 0.90. On the NTUH test set, the GAP model showed significantly better performance than the GMP model, with an AUROC of 0.65 versus 0.53, an AUPRC of 0.90 versus 0.88, an accuracy of 0.74 versus 0.50, and an F1 score of 0.84 versus 0.65.

These results demonstrate that the model using GMP performed reasonably well on

doi:10.6342/NTU202403906



the holdout test set but slightly underperformed compared to the GAP model. Specifically, while both models showed strong metrics on the PhysioNet test set, the GAP model had marginally better AUROC, accuracy, and F1 scores. However, a notable difference emerged when evaluating the models with the NTUH dataset. The GAP model maintained good generalization, achieving respectable scores across all metrics, while the GMP model showed a significant drop in performance, with the AUROC score indicating a performance close to random chance.

This discrepancy suggests that the choice of pooling strategy has a substantial impact on the model's ability to generalize across different datasets. The findings support the current study's choice of utilizing GAP, as it computes the final embedding by averaging all token embeddings, thereby giving equal weight to all tokens. This approach can capture a more holistic representation of the input sequence. In contrast, GMP focuses only on the highest values in the embeddings, potentially neglecting relevant but less prominent features. As a result, GMP might be less robust in scenarios where important information is distributed across many tokens rather than concentrated in a few. The consistency in the GAP model's performance across diverse datasets underscores the importance of considering pooling strategies that maintain a comprehensive representation of the data, particularly in complex domains like EEG analysis.

doi:10.6342/NTU202403906

Table 4.5: NTUH dataset results when trained with 80% training set compared to the entire PhysioNet dataset.

Metric	80% Training Set	Entire Dataset
AUROC	0.65	0.73
AUPRC	0.90	0.93
Accuracy	0.74	0.70
F1 measure	0.84	0.80

4.5 Training with Entire PhysioNet Dataset

4.5.1 Evaluation with NTUH Dataset

In this next experiment, we compared the results when the model is trained with the entire publicly available PhysioNet dataset (607 patients) with the original setup of being trained with the 80% training set split (485 patients). It is intuitive that with even more data used for training, the model should perform better on the external NTUH dataset. Table 4.5 shows the results of both setups when evaluated with the NTUH dataset. Notably, the model trained with the entire dataset uses a different optimized prediction threshold at 0.62 compared to the threshold of 0.55 for the 80% split model.

These results show a significant increase in AUROC from 0.65 to 0.73 when evaluated with the model trained with the entire dataset. The AUPRC also had a slight increase from 0.90 to 0.93. Both accuracy and F1 measures decreased when evaluated with the entire dataset, but the difference wasn't very significant. Overall, the model trained with the entire dataset performed better than the external NTUH dataset. Figure 4.6 shows the ROC (left) and the PRC (right) curves when the model trained with the entire dataset was used to evaluate the NTUH dataset.

We investigated the amount of correctly and incorrectly predicted positive and neg-

doi:10.6342/NTU202403906

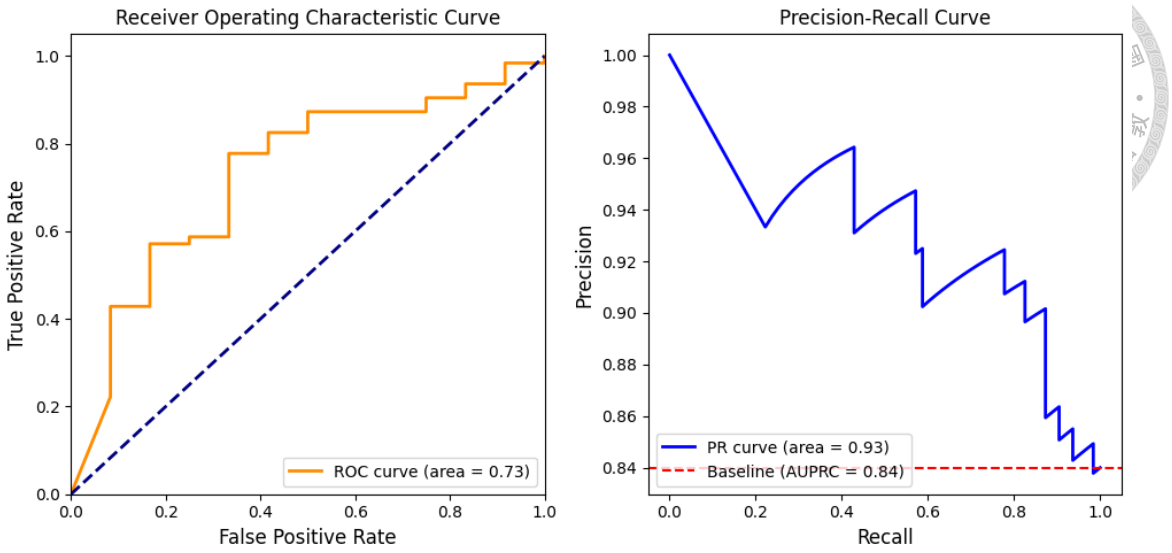


Figure 4.6: ROC (left) and PRC (right) curves when evaluated with the NTUH dataset using the model trained with the entire publicly available PhysioNet dataset.

ative samples through a confusion matrix, where positive samples are the class 1 samples or bad outcomes, while negative samples are the class 0 samples or good outcomes. The confusion matrix is a useful plot for analyzing the amount of TN (upper left), FN (upper right), FP (lower left), and TP (lower right). Figure 4.7 shows the confusion matrix obtained from the prediction results of the NTUH dataset when evaluated with the Entire Dataset model.

Through the confusion matrix, we can observe 8 true negatives and 4 false negatives, indicating that the model correctly predicted 8 out of 12 samples among the negative class samples, which shows an accuracy of 0.66 over the negatives. For the positive samples, we can observe 45 true positives and 18 false positives, indicating that the model correctly predicted 45 out of the 63 samples among the positive class samples, which shows an accuracy of 0.68 over the positives. These findings show that the model was able to distinguish the samples among the two classes very well despite the evident class imbalance, as it was able to predict more correct samples than incorrect samples in each class.

The sensitivity and specificity obtained from these results are 0.92 and 0.31, respec-

doi:10.6342/NTU202403906

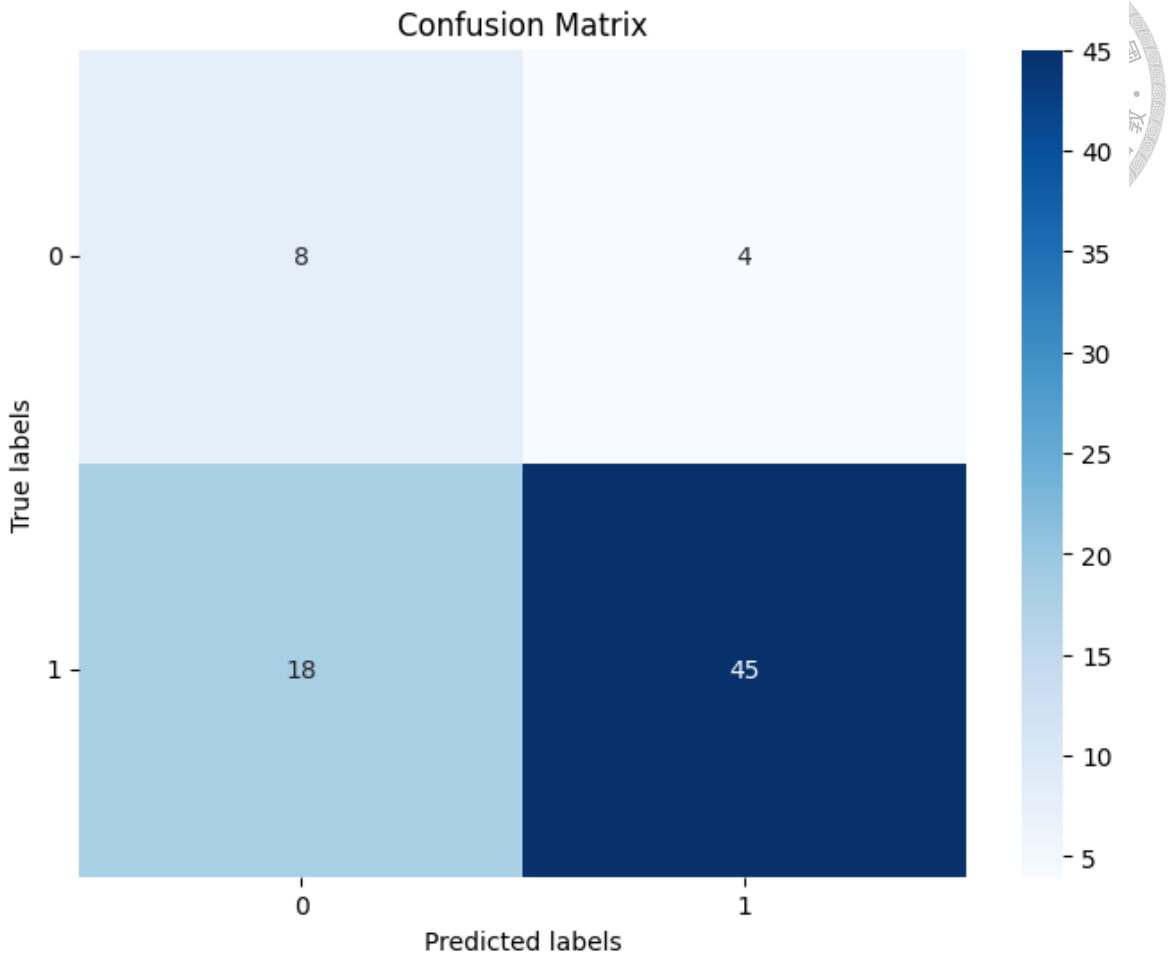


Figure 4.7: Confusion matrix from the prediction results of NTUH dataset when trained with the entire publicly available PhysioNet dataset.

tively. This disparity in metrics reflects the class imbalance present in the dataset, with a scarcity of negative cases. Due to this imbalance, specificity alone does not provide a comprehensive understanding of the model's performance on negative cases, as there are very few instances of these. However, the primary concern of this study is the accurate identification of positive cases or bad outcomes, as our goal is to minimize the number of patients incorrectly predicted to have bad outcomes. Therefore, sensitivity, which is 0.92, maybe a crucial metric in evaluating the final model's clinical interpretability. The high sensitivity indicates that the model is effective at identifying true positive cases, which is vital in this context.

doi:10.6342/NTU202403906

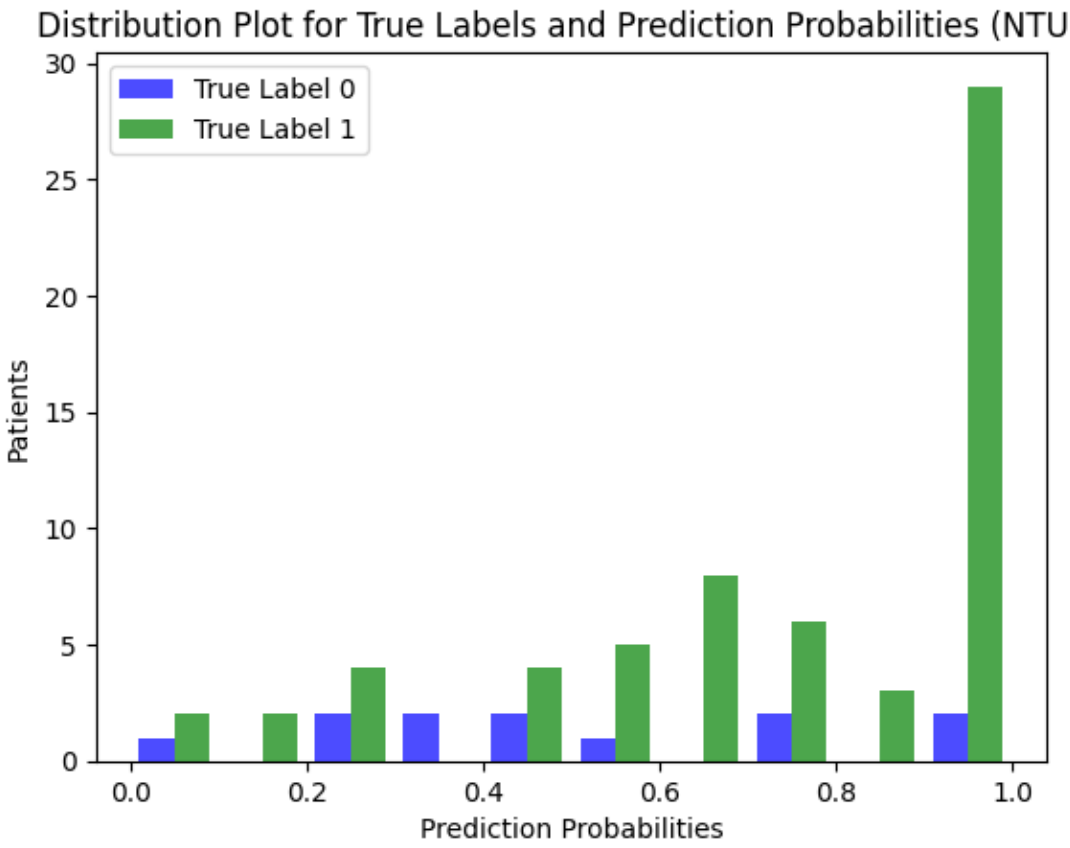
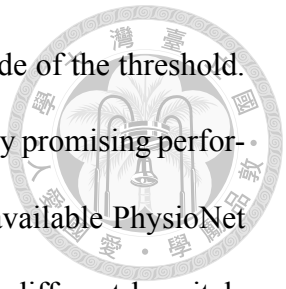


Figure 4.8: NTUH dataset distribution of prediction probabilities vs. true labels for the model trained with the entire PhysioNet dataset.

We plotted the prediction probabilities with the true labels from the model trained with the Entire Dataset when evaluated with the NTUH dataset to visualize how far or near the prediction probabilities are based on the true labels. Figure 4.8 shows the prediction probabilities and true labels when plotted in a single distribution. The green bars represent the true label 1s (bad outcomes), and the blue bars represent the true label 0s (good outcomes), while the x-axis represents the prediction probabilities and the y-axis represents the number of patients in the dataset.

We can observe from this distribution that the model, again, distinguishes the two classes very well, with most of the green bars beyond the threshold of 0.62 while most of the blue bars are before the threshold. The green bars are seen to be mostly clustered very near 1.0, indicating the model's confidence in predicting the positive classes correctly.

doi:10.6342/NTU202403906



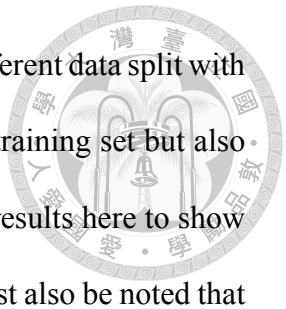
However, the blue bars are mostly evenly distributed along the left side of the threshold. Nevertheless, these findings demonstrate that our model achieves a very promising performance on an external dataset when trained using the entire publicly available PhysioNet dataset. This suggests potential future applications in datasets from different hospitals across the globe, as the model shows robust generalization capabilities across datasets from different backgrounds. Future studies could further explore deploying this model on entirely new datasets and conduct additional evaluations to determine whether its performance merely reflects some similarities between the NTUH dataset and the PhysioNet dataset or if it can genuinely generalize well when trained with the entire PhysioNet training set.

4.5.2 Baseline Comparisons

Finally, in this subsection, we compare the results of our model with other baselines, particularly from the challenge, to show how our model compares. We aim to show here whether attention mechanisms truly perform well with continuous EEG sequences through their segmented time steps by comparing them with the baselines.

Among the studies in the challenge that used an attention-based model [27–29], only our previous study [29] was able to have the proposed model evaluated through the challenge's official phase. Thus, the others [27, 28] were not used in this comparative analysis since their final results from the official phase are not available. Despite not using an attention-based model, the challenge winner [30] was used as a baseline as it is vital to show where our model stands with the challenge winner. A previously published study [13], despite not being part of the challenge, was also used as a baseline since they used the same dataset as was used in the challenge. It is important to note here that we

doi:10.6342/NTU202403906



cannot directly compare our results with their results as they used a different data split with their study, training their model not only with the publicly available training set but also with the hidden validation and test sets. However, we included their results here to show how our model compares despite the difference in the data split. It must also be noted that only Zheng et al. [13] used EEG as the sole input, while Zabihi et al. [30] used ECG with EEG, and our previous study [29] used clinical data with EEG to train the models.

The AUROC results of our proposed model, when evaluated with cross-validation through the training set, are plotted in Figure 4.9, together with the results from the benchmarks. Our model generally outperforms the other benchmarks when evaluated at earlier hours of 12 hours, 24 hours, and 48 hours. When evaluated at 72 hours, our proposed model outperforms Zheng et al.'s and our previous study's models while performing on par with the challenge winner's. Another interesting finding when evaluated at 12 hours is that the other benchmarks all performed at their worst when evaluated at this very early hour threshold. However, our model was already able to perform very well when evaluated at this hour. Although there was a slight decrease in performance in our proposed model when evaluated at 24 hours, the difference is not very significant, and the results at this hour still outperform even the challenge winners.

We note here that these benchmark comparisons are only meant to show how our proposed model's results stand with the previous studies, and readers must take precautions in making direct comparisons since each benchmark's setups differ. However, all studies cited in this benchmark comparison have the same focus on predicting neurological outcomes from coma patients and make use of the same publicly available dataset from the 2023 PhysioNet Challenge.

doi:10.6342/NTU202403906

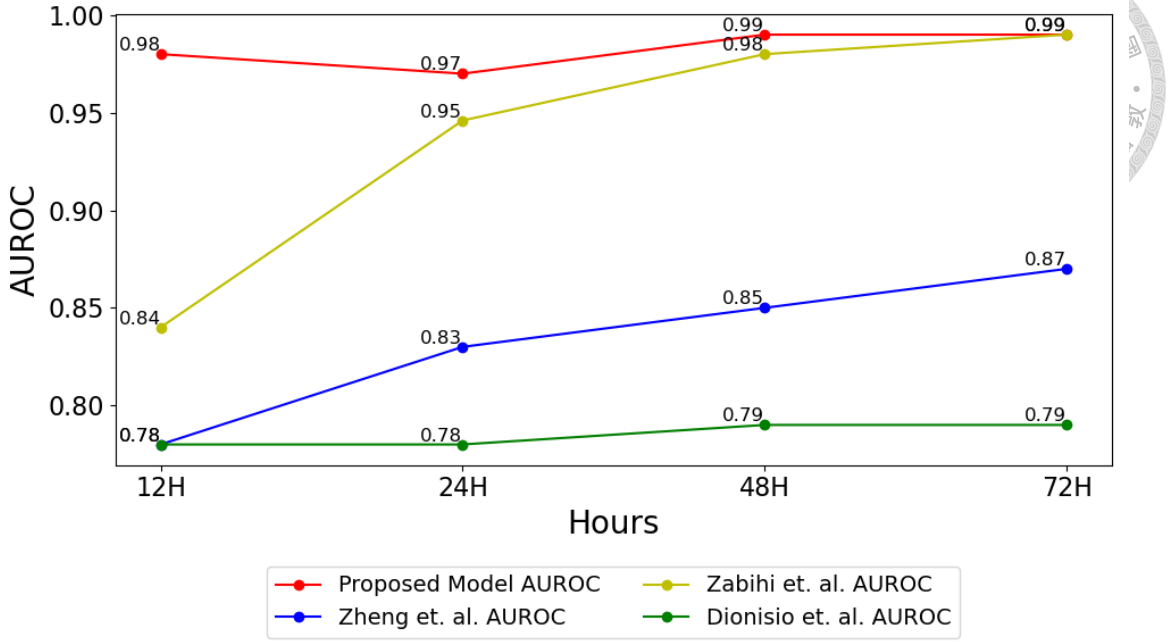


Figure 4.9: AUROC benchmark comparison across the entire PhysioNet public training set via cross-validation.

4.6 Visualizing the Model

In clinical scenarios, understanding why a model makes certain predictions is crucial for ensuring reliability and trust. Deep learning models, particularly those based on complex architectures, often act as ”black boxes,” making interpretability challenging. However, attention-based models, such as the Transformer, offer some level of insight into their decision-making processes.

By visualizing attention weights from the MHA layers, we can gain a clearer understanding of which features from the initial embedding are most influential in the computation of the learned output embeddings. This is done by analyzing both self-attention (how a token attends to itself) and cross-attention (how a token attends to other tokens).

In practice, features that exhibit high attention scores in these visualizations are considered more important, as they contribute significantly to the output embeddings. To

doi:10.6342/NTU202403906

make these visualizations more interpretable, we normalize the attention weights to a range between 0 and 1. This ensures that all weights are on a common scale, facilitating clearer comparisons.

We obtained the attention weights from the trained model's MHA layers and created correlation heatmaps to visualize both self-attention and cross-attention for each feature. These heatmaps help in understanding how each feature interacts with other features, providing valuable insights into the model's behavior.

In the first layer, as illustrated in Figure 4.10, the model primarily relies on earlier features to compute the output embedding. This is evident from the concentration of high and low attention scores among interactions between early features and both early and later features. This suggests that the initial layer focuses on leveraging the initial set of features more heavily.

In contrast, Figures 4.11 and 4.12 show that in the second and third layers, the model distributes attention more evenly across all features within each token. This indicates that the model increasingly utilizes a broader range of features when computing the output embedding, maximizing the use of all available data. The more distributed attention in these layers reflects the model's enhanced capability to integrate information from various features.

The fact that the third layer's attention patterns are similar to those of the second layer suggests that additional layers beyond the third may not have significantly improved performance. This could explain why adding more layers after the third did not result in notable gains in model performance. Essentially, the model may have already captured the necessary patterns and relationships in the data by the third layer, making further layers

doi:10.6342/NTU202403906

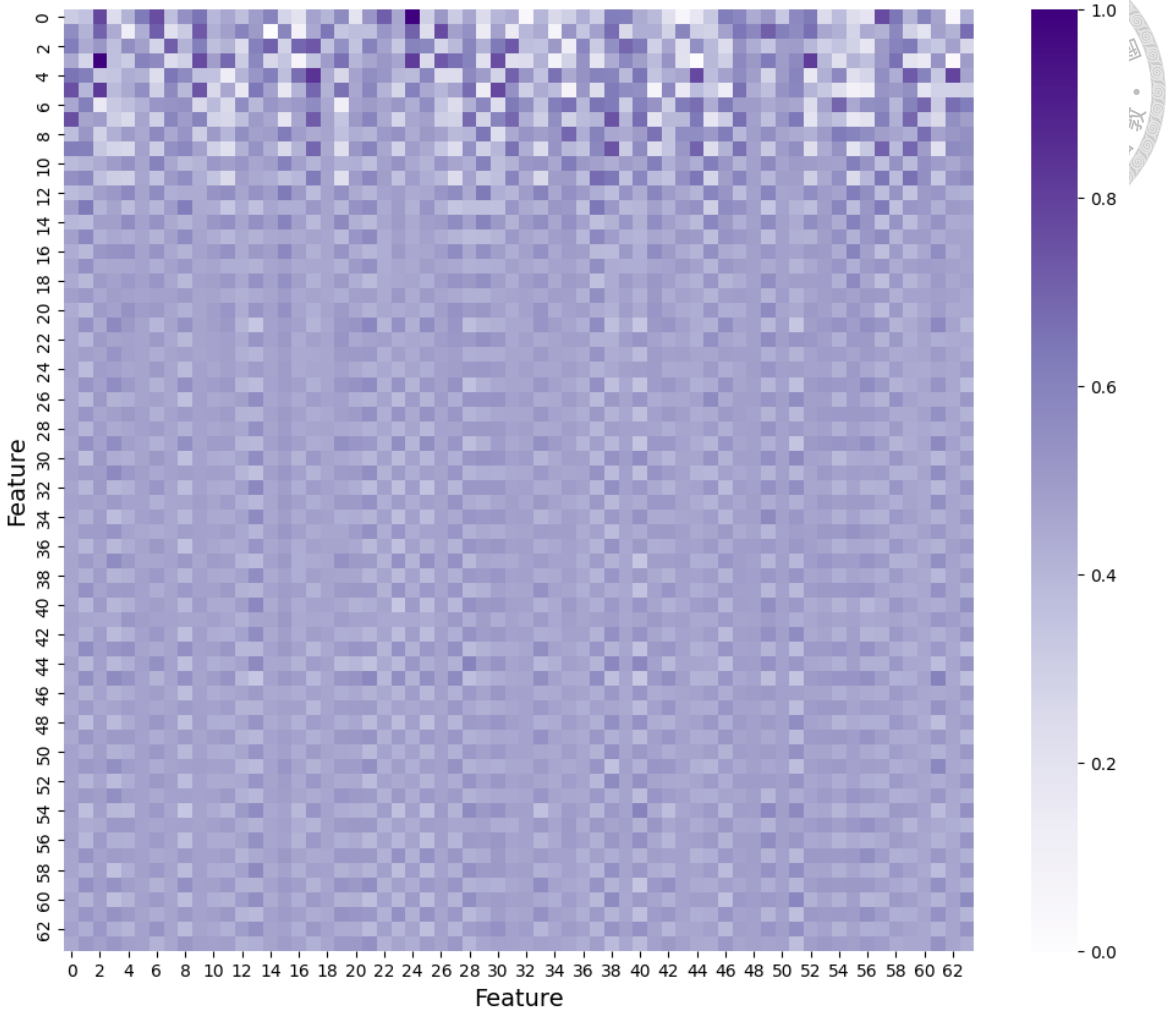


Figure 4.10: Attention map of first MHA layer.

redundant in terms of improving model performance.

The attention heatmaps reveal how different features interact with one another in the learned embedding space. By examining these interactions, we gain insight into which parts of the learned embedding are most influential in the model's decision-making process. This analysis allows us to trace back how the model weights various aspects of the input embedding.

Specifically, the attention heatmaps provide a visual representation of how each token's self-attention and cross-attention contribute to the final output. By identifying patterns in these attention weights, we can infer which embedding features are most critical

doi:10.6342/NTU202403906

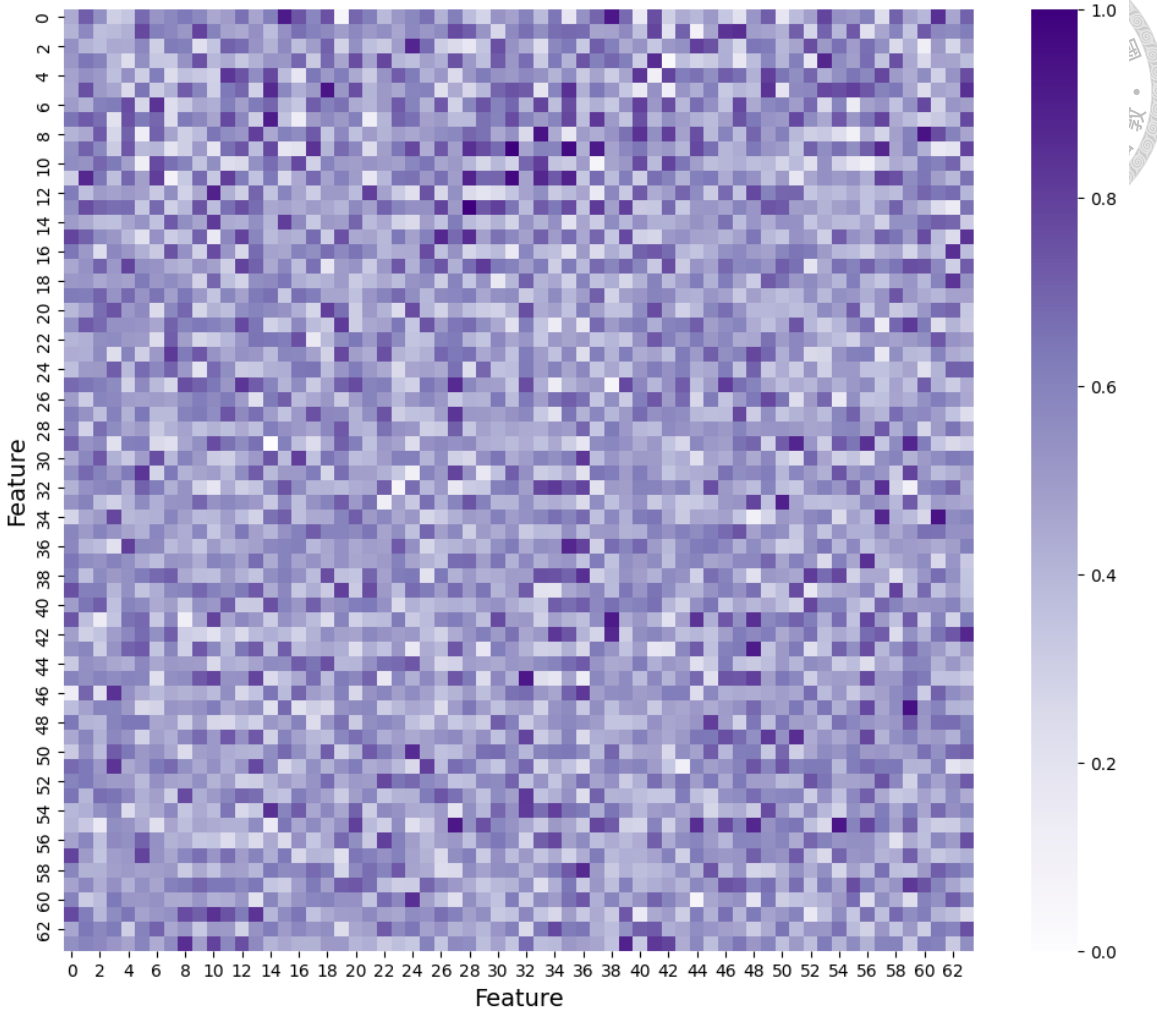


Figure 4.11: Attention map of second MHA layer.

for the predictions of the model. For instance, features with consistently high attention scores are likely to have a greater impact on the model's performance, whereas those with lower scores might be less influential.

To further evaluate the overall importance of each feature in the embedding space, we calculated the sum of attention weights for each feature after the final MHA layer. Figure 4.13 presents these summed attention weights, revealing a standardized distribution of feature importance. This distribution aligns with the previous attention map at the last layer, where all features contribute nearly equally to the model's predictive performance.

Among the features, Feature 38 showed the lowest contribution to the model, while

doi:10.6342/NTU202403906

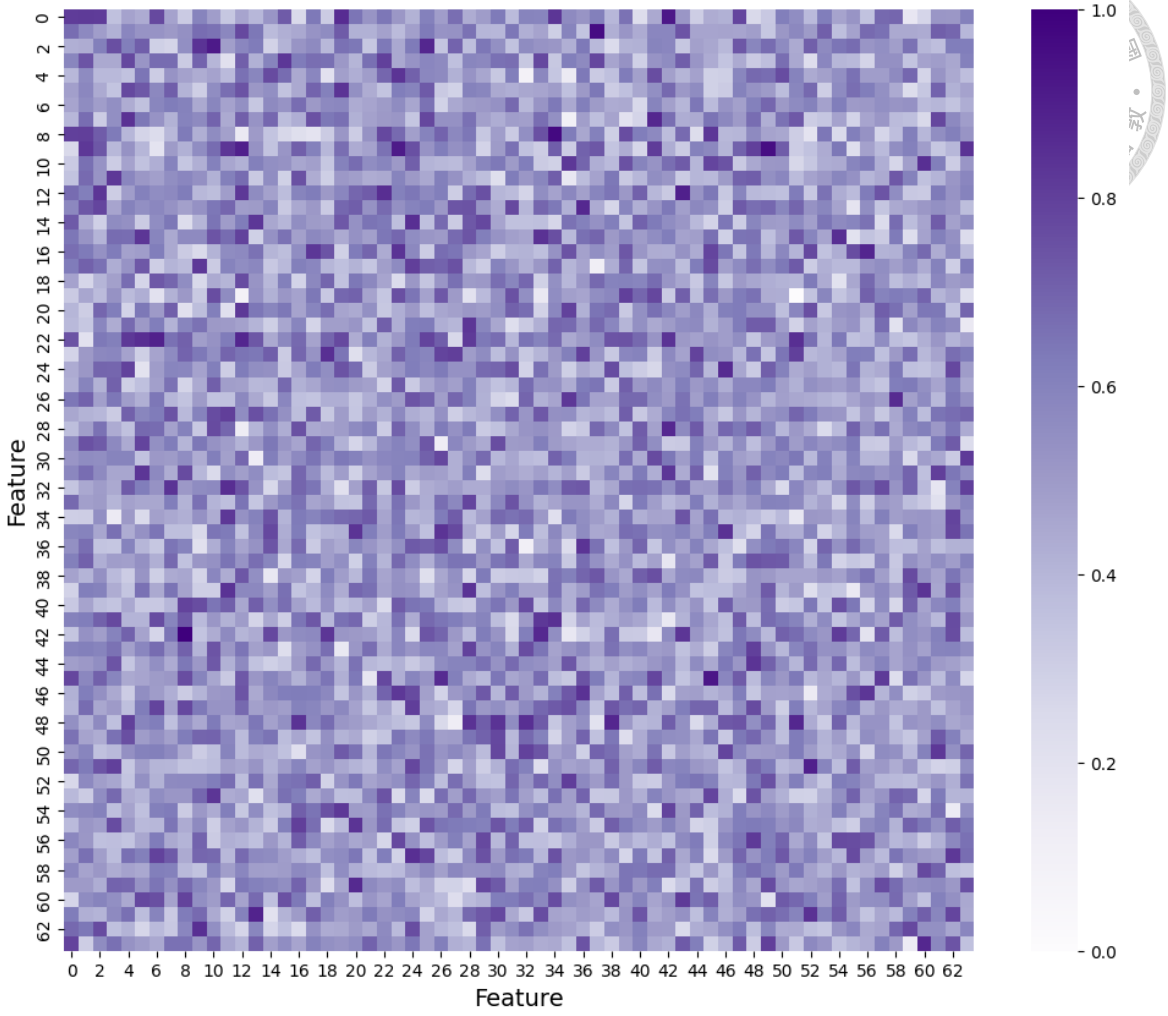


Figure 4.12: Attention map of third MHA layer.

Feature 22 had the highest contribution. This analysis highlights the relative significance of individual features within the learned embeddings, providing insight into which features from the initial embedding are most influential in the model's decision-making process.

Although Features 22 and 38 represent the extremes in terms of contribution, their importance is still relatively close to that of other features, suggesting that the differences are not stark. This indicates that these features are not literal outliers but part of a more evenly distributed significance spectrum. The summed attention weights across features reflect this balanced distribution, implying that the model does not heavily favor any particular feature. As a result, the model may exhibit less bias, considering a wide range of

doi:10.6342/NTU202403906

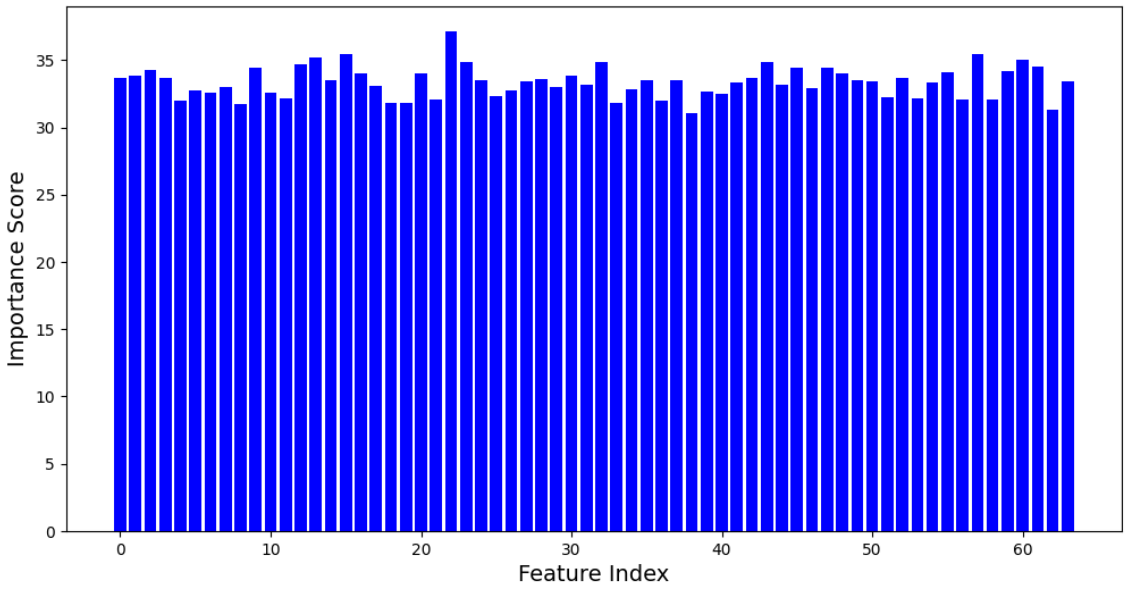


Figure 4.13: Feature importance distribution at the final MHA layer.

features in its decision-making process. This balance enhances the model's robustness and fairness, potentially improving its generalization and reliability in various scenarios.

It is important to note that these features in the learned embedding are combinations of all the original features resulting from the transformation through a linear, fully connected layer. Because each learned feature is a mixture of contributions from all original features, it is challenging to directly associate these learned features with specific real-world phenomena. Instead, the analysis focuses on understanding the model's internal dynamics rather than mapping learned features to specific real-world variables.

4.7 Analyzing Patient-wise EEG

Despite the insights gained from the learned embeddings, understanding the real-world significance of the original EEG features remains crucial. EEGs are often characterized by the patient from which they were recorded. To understand the similarities and differences in the characteristics of each patient's EEG recordings, we first aggregated the

doi:10.6342/NTU202403906

extracted EEG features into a single mean feature vector per patient. In Figure 4.14, we present the mean, median, standard deviation, and interquartile range (IQR) for all features across patients. By visually analyzing the distribution of these statistical measures, we can identify distinct outliers among the patients.

Upon analyzing the results, we found notable variations among the patients. Patient 24 had the highest mean score at 0.035, while patient 526 had the lowest at 1.7175e-05. For the median, patient 597 had the highest score at 0.028, and patient 100 had the lowest at 1.4898e-06. When looking at the standard deviation, patient 506 had the greatest variability with a score of 0.0364, whereas patient 526 had the least variability at 3.5280e-05. The IQR analysis revealed that patient 506 had the widest range at 0.0434, while patient 100 had the narrowest range at 2.8537e-06.

These patients are among the notable outliers, representing the maximum and minimum values among those identified. This pattern demonstrates that EEG characteristics are highly patient-dependent, reflecting unique physiological or neurological conditions. The significant differences observed among these patients underscore the importance of further studies to carefully integrate EEG data from different patients. Properly accounting for this variability is crucial to achieving accurate and reliable results, particularly in personalized treatment or in developing generalized models for broader applications.

4.8 Limitations

Our study poses some limitations that future studies may intend to tackle for further improvements. The first limitation of our study is its focus on training solely with EEG data, unlike previous studies in the challenge [27–30] that used a multimodal approach.

doi:10.6342/NTU202403906

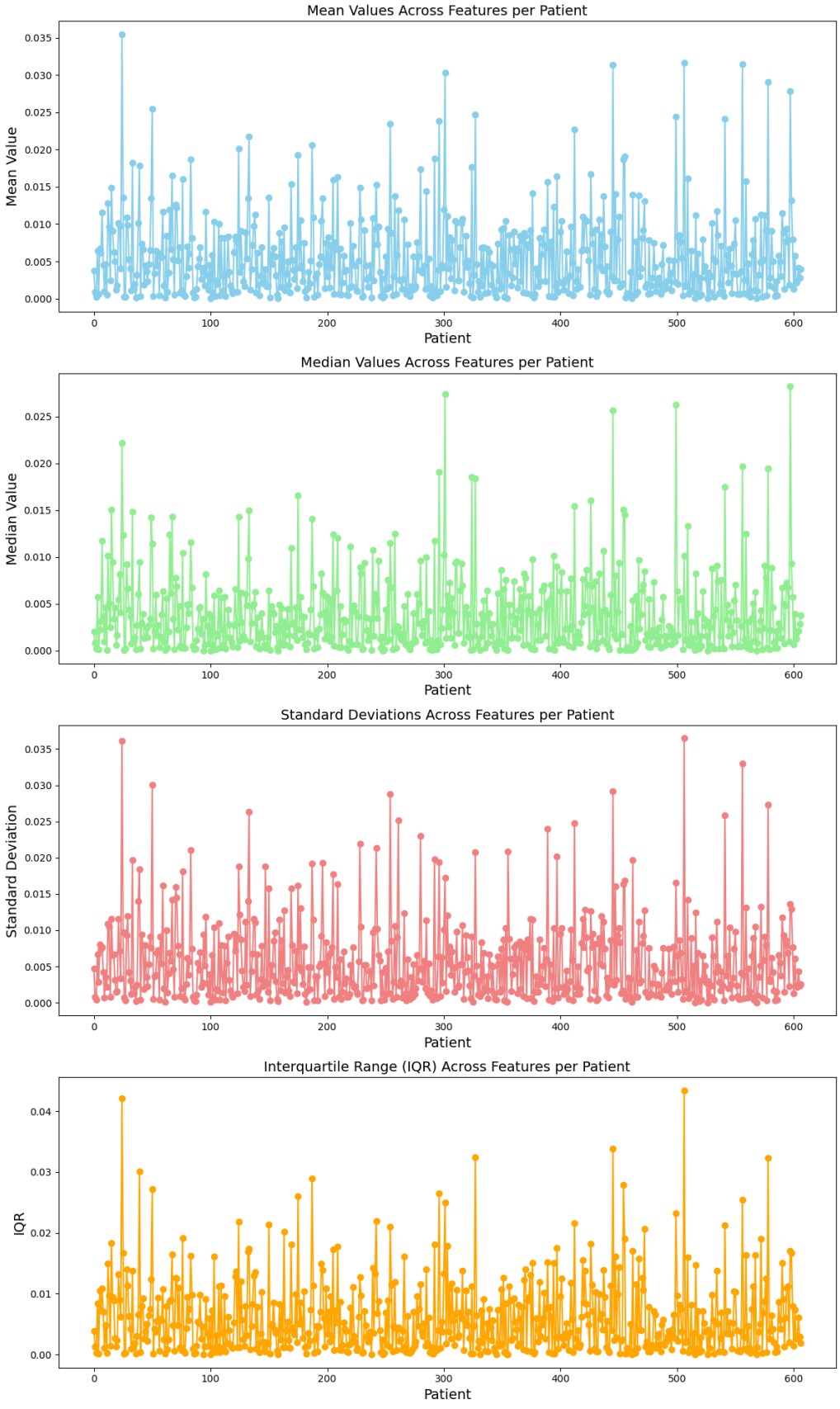
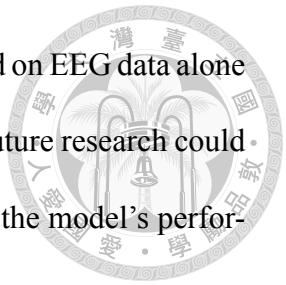


Figure 4.14: Statistics of each feature across all patients

doi:10.6342/NTU202403906



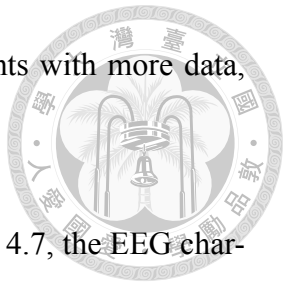
Our experiments demonstrated excellent results, with the model trained on EEG data alone outperforming benchmarks that utilized multimodal data. However, future research could extend our method to include multimodal data to potentially enhance the model's performance and generalizability.

Another limitation is the absence of more intensive channel selection among the EEG channels used. Some channels are known to be susceptible to external or biological artifacts. Traditional EEG processing often employs techniques such as principal component analysis (PCA) [59], independent component analysis (ICA) [60], and wavelet transform [61] to remove unwanted components from EEG as what previous studies [62–64] have proposed. Future studies may utilize such techniques to employ channel selections to potentially improve the results of the current method.

Furthermore, this study employed minimal signal processing steps, focusing on optimizing the Transformer model and evaluating whether attention mechanisms alone could identify patterns within the EEG data. Future research might explore more extensive signal processing pipelines to further eliminate biological artifacts, resulting in cleaner samples. However, excessive signal processing may distort EEG signals from their real-life clinical form, potentially leading to poorer prognoses. Future studies should be cautious not to overly remove data, as some might actually contribute valuable information to the model.

Another limitation of this study is the potential for model bias due to the varying number of EEG recordings per patient. While using recording-wise samples from all patients addressed the issue of class imbalance, it also increased the likelihood that the model learned more patterns from patients with a greater number of recordings. As shown in Figure 4.15, the distribution of EEG recordings per patient varies significantly. This im-

doi:10.6342/NTU202403906



balance may have caused the model to develop a bias toward patients with more data, potentially impacting the generalizability of the results.

Moreover, as highlighted by the statistical analysis from Section 4.7, the EEG characteristics are highly patient-dependent, with significant variability in measures such as mean, median, standard deviation, and IQR. The presence of outliers, particularly those patients exhibiting extremely high and low values, suggests that certain patients' unique patterns might disproportionately influence the model. This patient dependency further complicates the model's ability to generalize across a diverse population.

Our choice of selecting only the first 72 hours of recordings for each patient helped reduce the potential for overrepresentation of patients with more extensive EEG data. However, this approach may not entirely eliminate patient bias. Future research could further refine the methodology to ensure a more balanced representation of recordings per patient, thereby minimizing any residual bias and enhancing the generalizability of the model. Additionally, incorporating strategies to account for patient-specific variability, such as stratified sampling or advanced normalization techniques, could improve the robustness of the model and its applicability across diverse patient groups.

Finally, our study used PSD as an EEG feature. Future research could investigate other types of QEEG features to further enhance model performance. Another research direction may involve using raw EEG data to learn features through deep learning approaches. Some previous studies [65–68] have utilized CNN-based models to extract features from EEG, typically using short recordings from small datasets that require less computational power. However, this end-to-end setup may prove difficult with the dataset used in this study due to its extremely large size and the high dimensionality of each EEG

doi:10.6342/NTU202403906

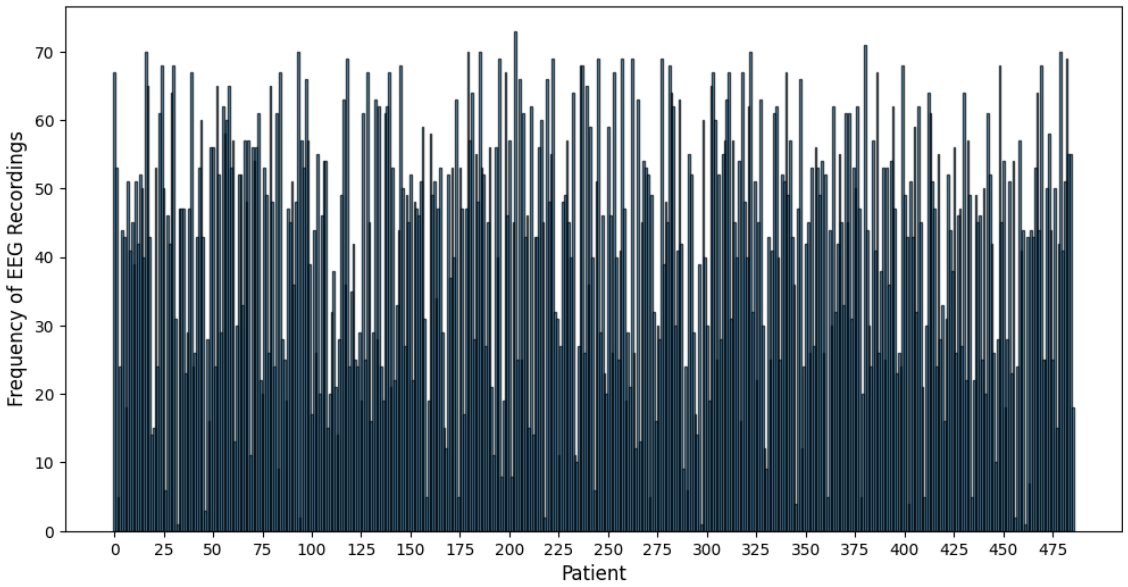


Figure 4.15: Distribution of EEG recordings per patient

sequence. With access to higher computational power, future research could explore learning direct feature embeddings from raw EEG and compare the results with models trained on extracted QEEG features.

doi:10.6342/NTU202403906

Chapter 5

Conclusion

Our findings in this thesis support our hypotheses and demonstrate that an attention-based model, specifically the Transformer, performs exceptionally well with EEG data when processed appropriately. The paragraphs below show our final evaluations of each of the objectives of the thesis.

1. **Recording-wise Training Method:** In comparing RW with PW model training, we validated our first hypothesis. The Transformer significantly improved when trained with RW EEG data because the attention mechanism benefits from the larger sample size. This allowed the model to learn better RW patterns and generalize well to any EEG recording in the test set, regardless of the hour it was recorded after ROSC.

2. **Capturing Long-Distance Temporal Patterns from Continuous EEG:** By comparing models trained on full hour-long EEG recordings to those trained on 5-minute subsampled epochs, we confirmed our second hypothesis. The Transformer can capture long-distance temporal patterns across hour-long EEG sequences more ef-

doi:10.6342/NTU202403906

fectively than the subsampled 5-minute epochs.

3. **Leveraging Only EEG Data to Train the Model is Enough:** Finally, our third hypothesis was confirmed by comparing our model to baseline methods. We found that EEG data alone is very capable of predicting neurological outcomes in comatose patients effectively since it was even able to outperform other models trained with multimodal data. The baseline comparison results also showed that our method of processing EEG data enables the Transformer to learn efficient long-distance temporal relationships among time-series tokens within each EEG sequence.

These findings show promising insights into understanding whether attention mechanisms work well with EEG sequences to accurately predict neurological outcomes in comatose cardiac arrest patients. Through this, we hope to aid physicians in making important clinical decisions since our model can achieve highly competitive results using an attention-based model over continuous time-series EEG.

doi:10.6342/NTU202403906

Ethics Statement

The private dataset used in this study was approved by the National Taiwan University Hospital Ethics Committee.

doi:10.6342/NTU202403906

Bibliography

[1] J. Engdahl, M. Holmberg, B. Karlson, R. Luepker, and J. Herlitz, "The epidemiology of out-of-hospital 'sudden' cardiac arrest," *Resuscitation*, vol. 52, no. 3, pp. 235–245, 2002.

[2] R. Geocadin, M. Buitrago, M. Torbey, N. Chandra-Strobos, M. Williams, and P. Kaplan, "Neurologic prognosis and withdrawal of life support after resuscitation from cardiac arrest," *Neurology*, vol. 67, no. 1, pp. 105–108, 2006.

[3] E. Jørgensen and S. Holm, "The natural course of neurological recovery following cardiopulmonary resuscitation," *Resuscitation*, vol. 36, no. 2, pp. 111–122, 1998.

[4] J. P. Nolan, C. Sandroni, B. W. Böttiger, A. Cariou, T. Cronberg, H. Friberg, C. Genbrugge, K. Haywood, G. Lilja, V. R. Moulaert *et al.*, "European resuscitation council and European society of intensive care medicine guidelines 2021: post-resuscitation care," *Resuscitation*, vol. 161, pp. 220–269, 2021.

[5] P. Sajda, A. Gerson, K.-R. Muller, B. Blankertz, and L. Parra, "A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 11, no. 2, pp. 184–185, 2003.

doi:10.6342/NTU202403906

[6] M. M. Ghassemi, B. E. Moody, L.-W. H. Lehman, C. Song, Q. Li, H. Sun, R. G. Mark, M. B. Westover, and G. D. Clifford, "You Snooze, You Win: the Physionet/Computing in Cardiology Challenge 2018," in *2018 Computing in Cardiology Conference (CinC)*, vol. 45, 2018, pp. 1–4.

[7] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, Physiotoolkit, and Physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

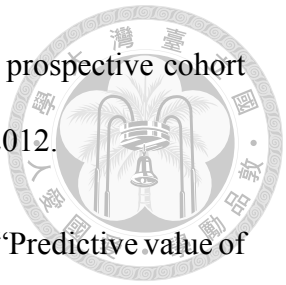
[8] M. A. Reyna, E. Amorim, R. Sameni, J. Weigle, A. Elola, A. B. Rad, S. Seyedi, H. Kwon, W.-L. Zheng, M. M. Ghassemi *et al.*, "Predicting neurological recovery from coma after cardiac arrest: The George B. Moody Physionet Challenge 2023," in *2023 Computing in Cardiology (CinC)*, vol. 50. IEEE, 2023, pp. 1–4.

[9] E. Amorim, W.-L. Zheng, M. M. Ghassemi, M. Aghaeeaval, P. Kandhare, V. Karukonda, J. W. Lee, S. T. Herman, A. Sivaraju, N. Gaspard *et al.*, "The international cardiac arrest research consortium electroencephalography database," *Critical Care Medicine*, vol. 51, no. 12, pp. 1802–1811, 2023.

[10] J. E. Wennervirta, M. J. Ermes, S. M. Tiainen, T. K. Salmi, M. S. Hynninen, M. O. Särkelä, M. J. Hynynen, U.-H. Stenman, H. E. Viertiö-Oja, K.-P. Saastamoinen *et al.*, "Hypothermia-treated cardiac arrest patients with good neurological outcome differ early in quantitative variables of EEG suppression and epileptiform activity," *Critical care medicine*, vol. 37, no. 8, pp. 2427–2435, 2009.

[11] M. C. Cloostermans, F. B. van Meulen, C. J. Eertman, H. W. Hom, and M. J. van Putten, "Continuous electroencephalography monitoring for early prediction of neu-

doi:10.6342/NTU202403906



rological outcome in postanoxic patients after cardiac arrest: a prospective cohort study," *Critical care medicine*, vol. 40, no. 10, pp. 2867–2875, 2012.

[12] L. R. Robinson, P. J. Micklesen, D. L. Tirschwell, and H. L. Lew, "Predictive value of somatosensory evoked potentials for awakening from coma," *Critical care medicine*, vol. 31, no. 3, pp. 960–967, 2003.

[13] W.-L. Zheng, E. Amorim, J. Jing, O. Wu, M. Ghassemi, J. W. Lee, A. Sivaraju, T. Pang, S. T. Herman, N. Gaspard *et al.*, "Predicting neurological outcome from electroencephalogram dynamics in comatose patients after cardiac arrest with deep learning," *IEEE transactions on biomedical engineering*, vol. 69, no. 5, pp. 1813–1825, 2021.

[14] Y. Du, Y. Xu, X. Wang, L. Liu, and P. Ma, "EEG temporal–spatial transformer for person identification," *Scientific Reports*, vol. 12, no. 1, p. 14378, 2022.

[15] J. Yan, J. Li, H. Xu, Y. Yu, and T. Xu, "Seizure prediction based on transformer using scalp electroencephalogram," *Applied Sciences*, vol. 12, no. 9, p. 4158, 2022.

[16] J.-Y. Guo, Q. Cai, J.-P. An, P.-Y. Chen, C. Ma, J.-H. Wan, and Z.-K. Gao, "A transformer based neural network for emotion recognition and visualizations of crucial EEG channels," *Physica A: Statistical Mechanics and its Applications*, vol. 603, p. 127700, 2022.

[17] M. Zeynali, H. Seyedarabi, and R. Afrouzian, "Classification of EEG signals using Transformer based deep learning and ensemble models," *Biomedical Signal Processing and Control*, vol. 86, p. 105130, 2023.

doi:10.6342/NTU202403906

[18] S. Lee, X. Zhao, K. A. Davis, A. A. Topjian, B. Litt, and N. S. Abend, "Quantitative EEG predicts outcomes in children after cardiac arrest," *Neurology*, vol. 92, no. 20, pp. e2329–e2338, 2019.

[19] S. Butterworth *et al.*, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[20] M. C. Tjepkema-Cloostermans, C. da Silva Lourenço, B. J. Ruijter, S. C. Tromp, G. Drost, F. H. Kornips, A. Beishuizen, F. H. Bosch, J. Hofmeijer, and M. J. van Putten, "Outcome prediction in postanoxic coma with deep learning," *Critical care medicine*, vol. 47, no. 10, pp. 1424–1432, 2019.

[21] S. B. Nagaraj, M. C. Tjepkema-Cloostermans, B. J. Ruijter, J. Hofmeijer, and M. J. van Putten, "The revised Cerebral Recovery Index improves predictions of neurological outcome after cardiac arrest," *Clinical neurophysiology*, vol. 129, no. 12, pp. 2557–2566, 2018.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[23] H. Wu, K. Meng, D. Fan, Z. Zhang, and Q. Liu, "Multistep short-term wind speed forecasting using transformer," *Energy*, vol. 261, p. 125231, 2022.

[24] W. Lyu, X. Dong, R. Wong, S. Zheng, K. Abell-Hart, F. Wang, and C. Chen, "A multimodal transformer: Fusing clinical notes with structured EHR data for interpretable in-hospital mortality prediction," in *AMIA Annual Symposium Proceedings*, vol. 2022. American Medical Informatics Association, 2022, p. 719.

doi:10.6342/NTU202403906

[25] G. Siddhad, A. Gupta, D. P. Dogra, and P. P. Roy, "Efficacy of transformer networks for classification of EEG data," *Biomedical Signal Processing and Control*, vol. 87, p. 105488, 2024.

[26] J. Liu, H. Wu, L. Zhang, and Y. Zhao, "Spatial-temporal transformers for EEG emotion recognition," in *Proceedings of the 6th International Conference on Advances in Artificial Intelligence*, 2022, pp. 116–120.

[27] M. Rohr, T. Schilke, L. Willems, C. Reich, S. Dill, G. Güney, and C. H. Antink, "Transformer Network with Time Prior for Predicting Clinical Outcome from EEG of Cardiac Arrest Patients," in *2023 Computing in Cardiology (CinC)*, vol. 50. IEEE, 2023, pp. 1–4.

[28] J. Pavlus, K. Pijackova, Z. Koscova, R. Smisek, I. Viscor, V. Travnicek, P. Nejedly, and F. Plesinger, "Using Embedding Extractor and Transformer Encoder for Predicting Neurological Recovery from Coma After Cardiac Arrest," in *2023 Computing in Cardiology (CinC)*, vol. 50. IEEE, 2023, pp. 1–4.

[29] J. Dionisio, C. Lin, L.-Y. Lin, and W.-C. Wu, "Predicting Neurological Outcomes of Comatose Cardiac Arrest Patients Using Transformer Neural Networks with EEG Data," in *2023 Computing in Cardiology (CinC)*, vol. 50. IEEE, 2023, pp. 1–4.

[30] M. Zabihi, A. C. Zar, P. Grover, and E. S. Rosenthal, "Hyperensemble learning from multimodal biosignals to robustly predict functional outcome after cardiac arrest," in *2023 Computing in Cardiology (CinC)*, vol. 50. IEEE, 2023, pp. 1–4.

[31] M. N. Alam, M. I. Ibrahimy, and S. M. A. Motakabber, "Feature extraction of eeg signal by power spectral density for motor imagery based bci," in *2021 8th Interna-*

doi:10.6342/NTU202403906

tional Conference on Computer and Communication Engineering (ICCCE), 2021, pp. 234–237.

[32] C. Kim, J. Sun, D. Liu, Q. Wang, and S. Paek, "An effective feature extraction method by power spectral density of eeg signal for 2-class motor imagery-based bci," *Medical & Biological Engineering & Computing*, vol. 56, no. 9, p. 1645—1658, Mar. 2018.

[33] R. Wang, J. Wang, H. Yu, X. Wei, C. Yang, and B. Deng, "Power spectral density and coherence analysis of alzheimer's eeg," *Cognitive Neurodynamics*, vol. 9, no. 3, p. 291—304, Dec. 2014.

[34] O. Dressler, G. Schneider, G. Stockmanns, and E. F. Kochs, "Awareness and the EEG power spectrum: analysis of frequencies," *BJA: British Journal of Anaesthesia*, vol. 93, no. 6, pp. 806–809, 09 2004.

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[37] S. J. Prince, *Understanding Deep Learning*. The MIT Press, 2023.

[38] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807—814.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

doi:10.6342/NTU202403906

[40] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456.

[41] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[44] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[45] K. Boyd, K. H. Eng, and C. D. Page, "Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals," in *Machine Learning and Knowledge Discovery in Databases*, H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 451–466.

[46] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal Thresholding of Classifiers to Maximize F1 Measure," in *Machine Learning and Knowledge Discovery in Databases*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 225–239.

doi:10.6342/NTU202403906

[47] B. Jennett and M. Bond, "Assessment of outcome after severe brain damage: a practical scale," *The Lancet*, vol. 305, no. 7905, pp. 480–484, 1975.

[48] L.-T. Ho, B. M. F. Serafico, C.-E. Hsu, Z.-W. Chen, T.-Y. Lin, C. Lin, L.-Y. Lin, M.-T. Lo, and K.-L. Chien, "Preserved Electroencephalogram Power and Global Synchronization Predict Better Neurological Outcome in Sudden Cardiac Arrest Survivors," *Frontiers in Physiology*, vol. 13, p. 866844, 2022.

[49] J. Hofmeijer, T. M. Beernink, F. H. Bosch, A. Beishuizen, M. C. Tjepkema-Cloostermans, and M. J. van Putten, "Early EEG contributes to multimodal outcome prediction of postanoxic coma," *Neurology*, vol. 85, no. 2, pp. 137–143, 2015.

[50] B. J. Ruijter, M. C. Tjepkema-Cloostermans, S. C. Tromp, W. M. van den Bergh, N. A. Foudraine, F. H. Kornips, G. Drost, E. Scholten, F. H. Bosch, A. Beishuizen *et al.*, "Early electroencephalography for outcome prediction of postanoxic coma: a prospective cohort study," *Annals of neurology*, vol. 86, no. 2, pp. 203–214, 2019.

[51] L. Sondag, B. J. Ruijter, M. C. Tjepkema-Cloostermans, A. Beishuizen, F. H. Bosch, J. A. van Til, M. J. van Putten, and J. Hofmeijer, "Early EEG for outcome prediction of postanoxic coma: prospective cohort study with cost-minimization analysis," *Critical care*, vol. 21, pp. 1–8, 2017.

[52] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. S. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, no. 267, pp. 1–13, 2013.

[53] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & systems (2nd ed.)*. USA: Prentice-Hall, Inc., 1996.

doi:10.6342/NTU202403906

[54] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[55] P. Welch, “The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,” *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[56] D. Yao, “A method to standardize a reference of scalp EEG recordings to a point at infinity,” *Physiological Measurement*, vol. 22, no. 4, p. 693─711, Oct. 2001.

[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[58] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[59] A. Maćkiewicz and W. Ratajczak, “Principal components analysis (PCA),” *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

doi:10.6342/NTU202403906

[60] P. Comon, “Independent component analysis, A new concept?” *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994, higher Order Statistics.

[61] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[62] Y. Zou, V. Nathan, and R. Jafari, “Automatic Identification of Artifact-Related Independent Components for Artifact Removal in EEG Recordings,” *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 73–81, 2016.

[63] I. Winkler, S. Haufe, and M. Tangermann, “Automatic Classification of Artifactual ICA-Components for Artifact Removal in EEG Signals,” *Behavioral and Brain Functions*, vol. 7, no. 1, p. 30, 2011.

[64] J. Gao, Y. Yang, J. Sun, and G. Yu, “Automatic Removal of Various Artifacts From EEG Signals Using Combined Methods,” *Journal of Clinical Neurophysiology*, vol. 27, no. 5, p. 312—320, Oct. 2010.

[65] Y. Ma, Y. Song, and F. Gao, “A novel hybrid CNN-Transformer model for EEG Motor Imagery classification,” in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8.

[66] H. Dose, J. S. Møller, H. K. Iversen, and S. Puthusserypady, “An end-to-end deep learning approach to MI-EEG signal classification for bcis,” *Expert Systems with Applications*, vol. 114, p. 532—542, Dec. 2018.

[67] Z. Wan, M. Li, S. Liu, J. Huang, H. Tan, and W. Duan, “EEGformer: A transformer—based brain activity classification method using EEG signal,” *Frontiers in Neuroscience*, vol. 17, Mar. 2023.

doi:10.6342/NTU202403906

[68] W. Y. Peh, P. Thangavel, Y. Yao, J. Thomas, Y.-L. Tan, and J. Dauwels, "Six-Center Assessment of CNN-Transformer with Belief Matching Loss for Patient-Independent Seizure Detection in EEG," *International Journal of Neural Systems*, vol. 33, no. 03, p. 2350012, 2023.

doi:10.6342/NTU202403906