2023

ENHANCING THE CLASSIFICATION OF AUTISM SPECTRUM DISORDER FROM RS-FMRI FUNCTIONAL CONNECTIVITY DATA USING TEMPORAL INFORMATION

Mihir Yashwant Ingole

Recommended Citation

Ingole, Mihir Yashwant, "ENHANCING THE CLASSIFICATION OF AUTISM SPECTRUM DISORDER FROM RS-FMRI FUNCTIONAL CONNECTIVITY DATA USING TEMPORAL INFORMATION" (2023). *Computer Science and Engineering Theses*. 519.
https://mavmatrix.uta.edu/cse_theses/519

ENHANCING THE CLASSIFICATION OF AUTISM SPECTRUM DISORDER FROM RS-FMRI FUNCTIONAL CONNECTIVITY DATA USING TEMPORAL INFORMATION

by

MIHIR YASHWANT INGOLE

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2023

To my parents and grandparents

ACKNOWLEDGEMENTS

Firstly, I would like to extend my deepest gratitude to my supervising professor Dr. Manfred Huber, under whose invaluable guidance I have been able to achieve this milestone. His exceptional insights and approach to problems has always captivated and significantly influenced my thinking. I regard this opportunity to do research under his guidance to be an extraordinary privilege.

Further I would like to thank Prof. Farhad Kamangar and Prof. David Levine for dedicating their valuable time to serve on my defense committee. Their involvement is greatly appreciated and has played a crucial role is the success of my defense.

Next, I am immensely grateful to my manager and mentor, Mr. Bito Irie, for providing me an opportunity to be part of CSE Services group and learn its values. His constant guidance and motivation has greatly inspired my thinking and, his support has been essential for my graduate studies at the University of Texas at Arlington.

None of this would have been possible without my mother's and father's never ending support and belief in me and my decisions for which I am truly grateful.

Lastly, my brother, whose life's challenges have been a source of inspiration and the driving force behind this study.

November 16, 2023

ABSTRACT

ENHANCING THE CLASSIFICATION OF AUTISM SPECTRUM DISORDER FROM RS-FMRI FUNCTIONAL CONNECTIVITY DATA USING TEMPORAL INFORMATION

MIHIR YASHWANT INGOLE, M.S.

The University of Texas at Arlington, 2023

Supervising Professor: Manfred Huber

Autism Spectrum Disorder (ASD) affects the patient's cognitive development which leads to difficulties in social functioning, daily tasks, and independent living. This necessitates intervention at an early age to take preventive measures and provide vital care. Manual diagnosis methods like Autism Diagnostic Observation Schedule (ADOS) assessment adopts symptom-based criteria which typically manifest at a later age. To automate this process, correlations computed from BOLD (Blood Oxygen-level dependent) signals obtained through resting state functional magnetic resonance imaging (rs-fMRI) data of patients across sparse brain regions has been used recently as a measure of functional connectivity. The goal of this study is to investigate the effect of temporal patterns in rs-fMRI time-series data through functional connectivity for automated identification of ASD using a world-wide multisite dataset called Autism Brain Imaging Data Exchange (ABIDE). Our suggested 2-stage network consisting of i) Ensemble Convolutional Neural Network (CNN) for feature extraction from correlation matrices of multiple shorter windows of rs-fMRI time-series and ii)

Temporal Convolutional Network (TCN) for classification on the same data after integrating the temporal dimension, has shown improvements in identification of ASD versus typical controls. Examining the rs-fMRI time-series functional connectivity in segments has shown higher gain in classification suggesting presence of relevant features in smaller segments. Due to the limited availability of functional neuro-imaging data for examination using a deep learning architecture, our study also demonstrates tackling the overfitting problem using noise injection and data augmentation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Introduction

Autism Spectrum Disorder (ASD) is a neurological and developmental disorder that causes restricted interests, repetitive behaviors and affects communication, social interaction, and cognitive functioning. These characteristics can make day-to-day activities very challenging. According to Centers of Disease Control and Prevention, ASD manifests before a child turns 3 and can persist throughout their lifetime, though signs might lessen as time progresses. Some kids exhibit ASD indicators within their first year, while for others these signs might only emerge after 24 months. Symptoms appearing as early as 2 years of age have been observed to remain stable up to 9 years and therefore early intervention for diagnosis and treatment is necessary [2]. Health care professionals typically diagnose ASD based on behavioral and developmental evaluations, often by the age of 2. While routine checkups include general developmental screenings, children at higher risk undergo more in-depth assessments. However, the absence of definitive early biomarkers hinders preemptive detection and optimal treatment opportunities. Functional neuroimaging studies have shown functional connectivity to be a reliable biomarker that can identify high-risk populations to determine the need for further diagnostic testing. Measures of brain region connectivity, obtained from functional magnetic resonance imaging (fMRI) scans, taken during the early pre-symptomatic period at 6 months of age, have demonstrated consistent accuracy, specificity, and sensitivity compared to Behavioral, Genetic, Immune and Medical History biomarkers [3].

1.2 Resting-state functional MRI Synopsis

Resting-state functional magnetic resonance imaging measures changes in blood oxygenation and flow, known as BOLD (Blood Oxygen-level dependent) signals, that result from neural activity. fMRI produces poor signal-to-noise ratio for task-based paradigms and reliable interpretations require aggregating data over multiple patients. On the other hand, resting-state (rs-fMRI) has produced consistent results on neuropsychiatric disorders such as Alzheimer's and shown to correlate behavioral and emotional measures [4]. A popular study by Biswal et al demonstrates correlation of low frequency fluctuations that functionally related brain regions exhibit in the resting state [5]. Functional connectivity is influenced by task-related factors and is typically used for examining neural activities across neuroimaging research. The BOLD time series over voxels of brain regions is averaged and correlations are calculated to investigate the connectivity patterns.

1.3 Overview

The work in this thesis primarily focuses on the analysis of temporal aspects of rs-fMRI time-series functional connectivity data for identification of ASD. It introduces a novel two-stage network architecture that comprises an Ensemble Convolutional Neural Network (CNN) for feature extraction and a Temporal Convolutional Network (TCN) for classification, showing improvements in distinguishing ASD from typical controls. Chapter 2 provides an overview of related Convolutional Neural Network (CNN) and Temporal Convolutional Network (TCN) applications, examining existing models used in neuroimaging and their efficacy in analyzing functional connectivity. Chapter 3 then reviews various studies that employed Deep Learning architectures, especially using the ABIDE dataset, for ASD identification through rs-fMRI BOLD functional connectivity. In Chapter 4, the composition, preprocessing and challenges of the ABIDE dataset are detailed. The following chapter outlines the methodologies employed in the study, specifically the two-stage architecture. This includes the preparation of fMRI time-series data, which involves handling variability in data length by slicing it with a constant window length. The CNN Ensemble Component architecture, using stacked CNN layers with varying kernel sizes for feature extraction, is also discussed. This chapter also delves into the 3D-CNN Temporal Convolutional Network (TCN) used for extracting temporal features for classification. Chapter 6 tackles the issue of overfitting occuring due to the complexity of connectivity matrices and the scarcity of data points. It explores the methods proposed in the thesis to combat overfitting, such as data augmentation through time-series segmentation and adding noise to datasets. Employing an autoencoder for regularization is also discussed with its limitations. Finally, Chapter 7 presents the analysis of the results from the Ensemble Component CNN (Stage 1), the Ensemble Component CNN on full-length time-series, and the 3D-CNN TCN (Stage 2). The TCN architecture outperformed the Stage 1 model and the Stage 1 model trained on full-length time-series. An experiment was conducted by shuffling the functional connectivity sequence to verify if classification improvements were attributed to temporal sequence information. The results suggested indicators that are present only for shorter segments and are averaged out when calculating the correlation matrix over the entire time series. Chapter 8 concludes the thesis and discusses potential future extensions.

CHAPTER 2

BACKGROUND

2.1 Convolutional Neural Network

In recent years, CNN has gained popularity in the field of Computer Vision, Natural Language Processing, Speech Recognition and Time-series Forecasting. Due to its advantages in weight sharing, dimensionality reduction and ability to create local connections, it has been widely used in industry and research for classification, detection, and retrieval related tasks. CNN is a feedforward hierarchical multilayered network with its core components consisting of convolutional layers, activation function/non-linear processing units, pooling layers/sub-sampling layers and fully connected/dense layers. The convolution kernels are useful in extracting features across the data points which are locally co-dependent. The output of the convolution kernels is passed through an activation function which introduces non-linearity and captures abstractions from the feature space. Down sampling is carried out after the output of the non-linear activation function which helps in reducing to only required/important features. The most popular ability of CNN is to exploit the spatial and temporal features of the input. The elements or pixels in the input are highly correlated and not independent of each other. CNNs leverage the property of convolutional filters to capture these spatial dependencies across elements of data points. This co-dependency can also be extracted from the temporal dimension of 3D inputs like videos or time-series fMRI scans.

Recent studies have also shown the use of CNNs in the field of neuroimaging and analysis of Autism Spectrum Disorder detection through functional connectivity and fMRI data. A study by Meszlényi, Buza, and Vidnyánszky [6] has introduced connectome-CNN to analyze brain connectivity matrices and use a subject's connectivity fingerprint for classification. Resting state fMRI data was used to obtain the BOLD signals from 499 brain Regions of Interest (ROIs) and further to calculate 499x499 connectivity matrices. 2 convolutional layers and then filters of size 1x499 for rows of the matrix and 499x1 filter size for remaining column vectors were designed for functional connectome. These were used to evaluate the data with Gaussian noise and have been shown to be useful for different functional connectivity measures like functional or structural information. Another study [7] showed the use of CNN for a multi-class classification task of classifying between ASD, Asperger's disorder (APD) and pervasive developmental disorder not otherwise specified (PDD-NOS). The authors in this paper used rs-fMRI data to obtain functional BOLD signals from 166 brain nodes. They selected a top-ranked node based-on one way analysis of variance test (ANOVA) of the power spectral density values (PSD). Scalograms of wavelet coherence of the top-ranked node and the remaining nodes are used as the dynamic functional connectivity input for the CNN and showed the potential of the wavelet coherence technique to be a good measure of functional connectivity and its use for diagnosis of neuropsychiatric disorders. In [8], the researchers introduced a CNN model with a combination of prototype learning (PL) and transfer learning, namely (T-CNNPL), to classify ASD through functional connectome. BOLD signals from 200 brain ROIs were used to calculate the functional connectivity matrix along with Kendall rank correlation to generate the input feature map. Transfer learning from non-medical to the medical image domain was achieved through fine-tuning the network layers and the fine-tuned CNN network was used as a feature extractor. These features were used for prototype learning where prototypes were calculated using Euclidean distance and a prototype loss based on distance cross-entropy was introduced for this model. CNNs have shown efficiency for diagnosis of ASD on non-functional connectome data. Authors of [9] have demonstrated a muti-channel CNN for early diagnosis of ASD on T1-weighted magnetic resonance images. T1w-MRI Scans of a total of 276 subjects fetched from the National Database for Autism Research (NDAR), initially 4 times skewed towards healthy patients, were used for evaluation. The authors introduced patch level data expanding where landmarks were selected using a data-driven landmark discovery algorithm to identify statistically significant landmarks and patches of dimension 24x24x24 were extracted. To unskew the dataset, 4 patches from ASD subjects and 1 from Normal Controls were selected and further each patch was fed to the multi-channel CNN which also utilized the global features like gender and volume for classification. [10] shows the use of CNNs to tackle the challenge of classifying high-dimensional dynamic functional connectivity (DFC) with limited samples. It introduces a specialized sparse convolutional neural network (SCNN) that uses a 1D-CNN backbone with three sparsity strategies to reduce overfitting, namely element-wise filter, 1x1 convolutional filter, and a sparse optimization classifier. The SCNN suggested has shown to extract relevant spatio-temporal features from DFC data and reasonable classification performance. CNNs have also demonstrated their ability to generate functional connectivity from time-series data of brain ROIs. [11] have presented a CNN, evaluated on the ADHD-200 dataset with a support vector machine, which extracts features from the BOLD signals of brain ROIs and a fully connected network that calculates the similarity between brain regions. [12] presents BrainNetCNN, a type of CNN which leverages the topological locality of brain connectome data, combined with global covariance pooling, which captures the second-order statistics of the features by calculating the covariance matrix of the feature maps. This has been used with a self-attention strategy to classify schizophrenia. In this review, we have examined the applications of Convolutional Neural Networks (CNNs) in neuropsychiatry, focusing on their effectiveness in processing functional connectivity and T1-weighted structural magnetic resonance imaging. We have also explored their utility in the context of neurodegenerative disorders such as Autism Spectrum Disorder (ASD), Auditory Processing Disorder (APD), Pervasive Developmental Disorder-Not Otherwise Specified (PDD-NOS), Attention Deficit Hyperactivity Disorder (ADHD), and schizophrenia.

2.2 Temporal Convolutional Network

Functional connectivity calculated through Pearson Correlation Coefficient (PCC) requires BOLD signals captured across brain ROIs to be averaged out for each ROI. This introduces a possibility of loss of information contained in the time-sequence. [13] compares the classification of ASD and TC, on 4D rs-fMRI data of the preprocessed ABIDE dataset by introducing four kinds on networks, namely i) CNN3D-TC which stacks temporal data in the input channel ii) CNN3D-MS which calculates mean and standard deviation across time points, iii) ConvGRU-CNN3D which uses a gated recurrent unit to process temporal information before spatial information, and iv) CNN4D which extends convolutional kernels to include the temporal dimension, integrating spatio-temporal processing throughout the network. The results show that the convGRU-CNN3D model yields the highest performance suggesting that explicit learning on the temporal dimension can enhance ASD classification from fMRI images. To explore the possibility of leveraging the temporal aspect, we have proposed the use of Temporal Convolutional Networks (TCN) on the time-series data. TCN has recently been introduced to the world of deep learning and is well establishing its ground in the field of sequence modeling. TCN models that combine best practices such as residual connections and dilated convolutions have been shown to substantially outperform recurrent architecture like GRUs and LSTMs and also exhibit longer memory for long-range information [14]. So far studies conducted for examining the temporal effects of the fMRI data during classification of ASD included networks trained on fMRI brain images for detecting temporal as well as spatial patterns. [15] presents a residual graph convolutional network where a residual attention network extracts both structural and temporal features of 4-dimensional fMRI brain images and ensures that the dynamic connectivity between two specific brain areas is maintained, and the graph convolutional network, which uses a connectivity matrix from 39 brain regions, uses the output of the residual network to construct a graph. The transformed graph learns the dynamic functional connectivity through a neural network and classifies the ASD and TC groups. Another study [16] explored the performance of 3D-CNN on rs-fMRI data of ABIDE-I and II datasets combined after summarizing the temporal dimension and maintaining full spatial resolution by applying various transformations. The performance after reducing the temporal dimension and applying 3D-CNN to the 3D spatial resolution found comparable results to a SVM for classification of ASD. [17] introduces Pseudo 4D ResNet designed to extract both temporal and spatial features from fMRI data efficiently. 3D spatial filters and a 1D temporal filter is used instead of a 4D spatiotemporal convolution to reduce computational costs. A data augmentation approach through sampling fMRI data at equal time intervals, similar to our work, is used. Main components of the network are the 4D maximum pooling layers to reduce dimensionality, P4D convolution blocks for initial feature extraction and mixed residual blocks to further extract the spatio-temporal features from the data. [18] proposes a deep learning framework for analyzing rs-fMRI by a two-stage process: first, using a convolutional autoencoder to minimize the noise presence in fMRI data temporally, and second, spatial correspondence across different individuals' brains is established through a convolutional neural network which identifies all relevant Resting State Networks (RSNs) and composites them into more complex RSNs for ASD classification.

CHAPTER 3

RELATED WORK

3.1 Studies on Functional Connectivity of ABIDE dataset

Recently, there have been a growing number of studies showing the use of machine learning and deep learning classifiers for analyzing rs-fMRI data. In [19], deep learning based rs-fMRI and structural MRI pipelines using LeNet and GoogleNet CNN models were employed. These methods successfully distinguished Alzheimer's patients and normal controls with an accuracy of 99.9% and 98.84%, respectively. [20] used functional connectivity of brain regions derived from rs-fMRI data and clinical information like age, gender, and genetic information of the subject for classifying AD (Alzheimer's Disease) and control patients. A 2-hidden-layer autoencoder was used with this data to achieve an accuracy of 86.47%.

[21] proposed another two-stage architecture to classify normal and schizophrenic patients. A 4-hidden layer autoencoder was used for feature extraction, trained in an unsupervised manner. The extracted features were subsequently used to train a binary Support Vector Machine (SVM) classifier, achieving 92% accuracy. [22] used fMRI to predict optimal deep brain stimulation (DBS) settings for Parkinson's disease patients. A machine learning model, trained on 67 PD patients' data, predicted the settings with 88% accuracy suggesting that fMRI and ML could enhance DBS adjustments in clinical settings. [23] introduced a Deep Neural Network (DNN) based subject-transfer decoder for brain decoding from fMRI data. Using principal sensitivity analysis (PSA) universal discriminative features were visualized, suggesting that

shared functional connectivity plays a pivotal role in characterizing task-specific brain activity.

The Autism Brain Imaging Data Exchange (ABIDE) is the most prevalent dataset for identification, prediction, and analysis of autism biomarkers through fMRI and machine learning techniques. It consolidates data from 17 distinct sites, offering 1,112 resting-state functional magnetic resonance imaging (rs-fMRI) scans, paired with structural MRI and phenotypic details, from 539 individuals with ASD and 573 typical controls [24]. [25] classifies ASD from control subjects using the ABIDE dataset, focusing on brain activation patterns. Features are extracted using the CC200 parcellation atlas, which identifies 200 brain regions for fMRI data collection. The mean time series from these regions are obtained for each participant. Correlations of these averages evaluate brain connectivity. The 200x200 correlation matrix yields 19,900 features; redundant diagonal and upper triangle values are removed. The analysis employs unsupervised pre-training with two stacked denoising autoencoders. The first autoencoder takes 19,900 features and compresses them to 1,000, while the second compresses 1,000 to 600. This data informs a Multilayer Perceptron with a 19,900-1,000-600-2 structure. Performance metrics include 70% accuracy, 74% sensitivity, and 63% specificity using leave-one-site-out cross-validation.

[26] introduces recurrent neural networks equipped with long short-term memory (LSTMs) for classifying ASD and TC. Rather than relying on measures of functional connectivity, this approach uses the ABIDE-I dataset's fMRI data directly for LSTM training and testing. Unlike traditional classification methods that analyze the entire time sequence before producing an output, this model classifies at every time point during input sequence processing making it more robust to the noisy fMRI data. The output from single nodes is averaged over the sequence and forwarded for classification, with a standard dropout layer added between the dense and pooling

layers. To ensure consistent sequence lengths across the dataset, input time courses are cropped and inputs per subject are increased, amplifying the dataset tenfold. For model validation, a 10-fold cross-validation is applied, maintaining subject proportions from each site. The dataset splits are 85% training, 10% testing, and 5% validation. Training halts if validation loss doesn't reduce over 20 epochs or after 300 epochs. The learned LSTM weights, when interpreted using the Neurosynth meta-analysis tool, reveal crucial anatomical regions significant for ASD abnormalities.

[27] introduces a graph-based classification system for the ABIDE dataset using a Deep Belief Network (DBN). The dataset, utilizing the CC200 functional parcellation, includes 200 ROI time series with differing timestamps based on the institution. From the 19,900 features obtained using the Pearson correlation coefficient (PCC), a two-step feature selection process is applied. The first step is the Graph Based Feature Selection (GBFS) using both external and internal measures. For external measures, connections must be significantly distinct in relation to autistic levels. This involves comparing the mean of each connection to the global mean, and ensuring the difference exceeds a threshold (set by α times the global standard deviation). This yielded 81 ASD and 380 TC connections. For internal measures, a community of ROIs is formed for nodes in each remarkable connection using the K-Nearest Neighbors graph extension, with K set to 6. This builds on the assumption that autistic neural activity might function collaboratively around a node rather than solely region-to-region. After filtering, 3,288 ASD connections remain, 16.52% of the total. The connections are then refined using a restricted path-based depth first search algorithm. The classification is done using a DBN model with automatic hyperparameter tuning. The DBN comprises three restricted Boltzmann machines and undergoes unsupervised pretraining followed by supervised fine-tuning. Hyper-

parameters are optimized using Bayesian optimization with Gaussian processes, and evaluation employs 10-fold cross validation.

In [28], a novel approach for ASD subject classification is explored by integrating data augmentation techniques with neural network models. The data, sourced from the ABIDE dataset using the CC200 parcellation atlas and CPAC pipeline, undergoes Pearson's correlation to determine functional connectivity from fMRI readings. Out of the 19,900 features derived, only a quarter of the extreme values are chosen for augmentation. The proposed data augmentation technique involves oversampling and synthetic data generation in feature space using k-Nearest Neighbors. The formula [$\mathrm{p}' = \alpha \times \mathrm{p} + (1 - \alpha) \times \mathrm{q}\ \mathrm{r}$] describes synthetic feature vector computation, where α ranges between 0.5 and 1. The Extended Frobenius Norm (EROS) aids in finding the nearest neighbor, computing similarities between two Multivariate time series based on their covariance matrices' eigenvalues and eigenvectors. Resulting synthetic samples double the training set. Feature size reduction employs an autoencoder, which outperforms PCA due to its non-linear dimension reduction. This autoencoder, alongside a single layer perceptron, is trained simultaneously, combining MSE reconstruction and binary cross entropy losses and obtaining 70.1% accuracy. This dual training aids in data reconstruction and retains classification-relevant information. The technique outshines many existing methods for intra-site datasets, with data augmentation boosting performance by 2%.

[29] utilized Deep Belief Networks (DBN) to analyze combined rs-fMRI and sMRI data from ABIDE I and ABIDE II datasets. The goal was to classify ASD from typical controls in subjects aged 5-10 years. Classification was based on the Automated Anatomical Labeling (AAL) parcellation atlas, which comprises 116 Regions of Interest (ROIs). The DBN is structured hierarchically with multiple Restricted Boltzmann Machines (RBMs). For compatibility with real-valued data, RBMs with

Bernoulli distributed units were used for visible layers, and binary outputs for hidden layers. The DBN's stacked RBM layers and employed logistic regression to compute classification probabilities in the final layer, facilitating error backpropagation and network fine-tuning. Tests were conducted using DBN depths of 2 and 3, and modalities combinations like rs-fMRI, Grey Matter (GM), White Matter (WM), and their various fusions. The accuracy of 65.56% was achieved by merging rs-fMRI, GM, and WM data, using a 10-fold cross-validation on a DBN of depth 3.

In [30], the authors employed multiple Support Vector Machines (SVMs) to differentiate between ASD and TC subjects using rs-fMRI data from the ABIDE dataset, comprising 46 TC and 61 ASD patients. Utilizing the AAL parcellation atlas, the fMRI image was segmented into 90 regions, with correlations determined via PCC. In the random SVM method, the dataset is split into training and test sets. Subsets of samples and features from the training set are used to build individual SVMs iteratively, forming a random SVM cluster. When testing, multiple SVMs simultaneously decide on a test subject, with majority decisions dictating the classification outcome. Given the unique feature selection for each SVM, their performances vary. The top 100 SVMs, based on performance, provide the feature matrix, with the 400 most frequent features labeled "important features". The random SVM cluster achieved an impressive 96.15% accuracy using 500 SVMs.

[31] presented a deep learning model named ASD-SAENet designed to classify ASD from typical controls using fMRI data. The model employs sparse autoencoders, optimizing feature extraction for classification purposes. These extracted features are inputted into a deep neural network to enhance the classification of fMRI scans associated with ASD. The CC200 parcellation method was adopted, producing 200 x 200 functional connections. Of these, 9,950 were chosen as inputs for each subject. A sparse autoencoder facilitated dimensionality reduction, capturing meaningful lower-

dimensional patterns. The autoencoder's bottleneck served as the input for the DNN, with unit sizes for successive layers being 4975, 2487, 500, and 2. Model assessment was conducted using 10-fold cross-validation on the entire dataset and 5-fold cross-validation per site, achieving an accuracy of 70.8%.

In [32], the authors applied traditional machine learning techniques to the ABIDE dataset. Previous applications of these algorithms lacked optimization in their parameters. To address this, the authors utilized logistic regression, ridge classifier, support vector machine (SVM) with an L2 penalty, and SVM with a Gaussian kernel, refining the parameters for each technique. The grid-search approach was employed for parameter optimization across seven brain atlases: CC400, CC200, AAL, HOA, TT, EZ, and Dosenbach. The highest accuracy achieved was 71.98% using the Ridge classifier on the CC400 parcellation atlas. Through the features chosen, the relationships between various brain regions were ascertained for both ASD and TC subjects.

[33] aimed to classify an individual with high-functioning autism using point-to-point functional connectivity derived from the ABIDE dataset. From the dataset's original 1,112 subjects across 17 sites, the study considered subjects that met the inclusion criteria, resulting in 517 typically developing individuals and 447 autism subjects from 16 sites. The study leveraged functional connectivity Magnetic Resonance Imaging (fc-MRI), given its ability to detect synchronized neural activity through slow BOLD signal fluctuations between brain regions. After preprocessing, mean time series were extracted from 7,266 gray matter Regions of Interest (ROIs). These averages were then correlated to analyze brain connectivity, forming a connectivity matrix. Each matrix cell contains a Fischer-transformed Pearson Correlation Coefficient, representing an association between ROIs. For each of the 26.4 million correlations, one was selected for all 964 subjects, and a general linear model was

fitted, factoring in age, age-squared, gender, and handedness. The estimated value for the omitted subject was adjusted for site variability and then compared with the actual value. This process was repeated for all connections, averaged over all ROIs, and summed to produce a classification score. A predominance of negative or positive scores indicated a control or autism subject, respectively.

The study in [1] delves into the automated detection of Autism Spectrum Disorder (ASD) by identifying potential biomarkers through the use of Convolutional Neural Networks (CNN). The research leverages resting-state functional Magnetic Resonance Imaging (rs-fMRI) data derived from the multi-site ABIDE dataset. From an initial set, 505 ASD patients and 530 controls were selected based on the specified inclusion criteria. Data was collected using the CC400 parcellation atlas, which segments the brain into 400 regions of interest (ROIs). The mean time series from these ROIs were extracted for each participant. By analyzing correlations between the averages of these ROIs, a connectivity matrix was constructed. Each matrix cell contained a Pearson Correlation Coefficient, which ranged from -1 to 1. For each subject, a 392x392 correlation matrix was derived using the CC400 atlas. This matrix was inputted into a CNN comprising seven stacked convolutional layers with 400 filters of dimensions from 1x392 to 7x392, based on the correlation matrix dimensions. Each convolutional layer was paired with a max-pooling layer and followed by densely connected layers. The combined layer outputs were fed to a Multi-Layer Perceptron (MLP) classifier with a 25% dropout rate. By assessing the gradient of class scores relative to the input image and deriving a saliency map, four critical brain areas significant for ASD diagnosis were identified. In a comparative analysis, the CNN model's performance surpassed traditional machine learning classifiers like SVM, KNN, and RF in accuracy, sensitivity, and specificity when evaluated using 10-fold cross-validation on the ABIDE-I dataset.

CHAPTER 4

DATASET AND PREPROCESSING

4.1 ABIDE Overview

Autism Brain Imaging Data Exchange I (ABIDE - I) is an aggregation of resting-state functional magnetic resonance imaging (rs-fMRI) datasets with corresponding structural MRI and phenotypic information from 17 different international sites with a total of 1,112 subjects out of which 539 are ASD and 573 typical control patients [24]. It basically contains T1-weighted images with phenotypic information which contains the demographic information, clinical measures, and other relevant information about the participants. The dataset is publicly available to researchers which has facilitated many studies and a wide range of analyses. One of its major challenges is its variability in data acquisition parameters and diagnostic procedures across sites. Phenotypic information in the ABIDE dataset consists of crucial demographic, clinical, and behavioral data, essential for understanding ASD's diverse presentations.

4.2 Preprocessing

For the purpose of this study, we used the pre-processed version of the ABIDE-I dataset provided by the Preprocessed Connectomes Projects (PCP) [34]. The dataset was downloaded using the nilearn python library with the Configurable Pipeline for the Analysis of Connectomes (C-PAC). The C-PAC's functional preprocessing initiative included slice time correction, motion correction, skull stripping, global mean intensity normalization and nuisance signal regression with 6 motion parameters, CompCor with 5 principal components [35], linear and quadratic drifts in fMRI signal. Data passing the quality assessments, and which employed global signal regression, bandpass filtering (between 0.01Hz and 0.1 Hz) were fetched with a total of 871 subjects (403: ASD and 468: TC) across 14 sites. Further the dataset was undersampled for each site resulting in a total of 788 subjects across 12 sites. This is required since the model selection after Stage 1 network is done on the basis of accuracy and unbalanced datasets can produce bias while model selection.

CC400 parcellation atlas was used to collect the mean time-series across 392 brain regions [36]. This is performed using a clustering method where the brain is segmented into regions such that voxels within a region have similar time-series profiles during rs-fMRI scans. The functional connectivity is calculated with Pearson correlation coefficient among the 392 brain regions producing 392 x 392 correlation matrix for each data point. Every cell in the matrix contains a value in the range -1 to 1, where -1 represents that the corresponding brain regions are anti-correlated and 1 represents highest correlation.

Fig. 4.1 illustrates the time-series of highest correlated brain regions of a subject with a correlation coefficient of 0.82457 and Fig. 4.2 illustrates the time-series of most negatively correlated brain regions of a subject with a correlation coefficient of -0.67955.

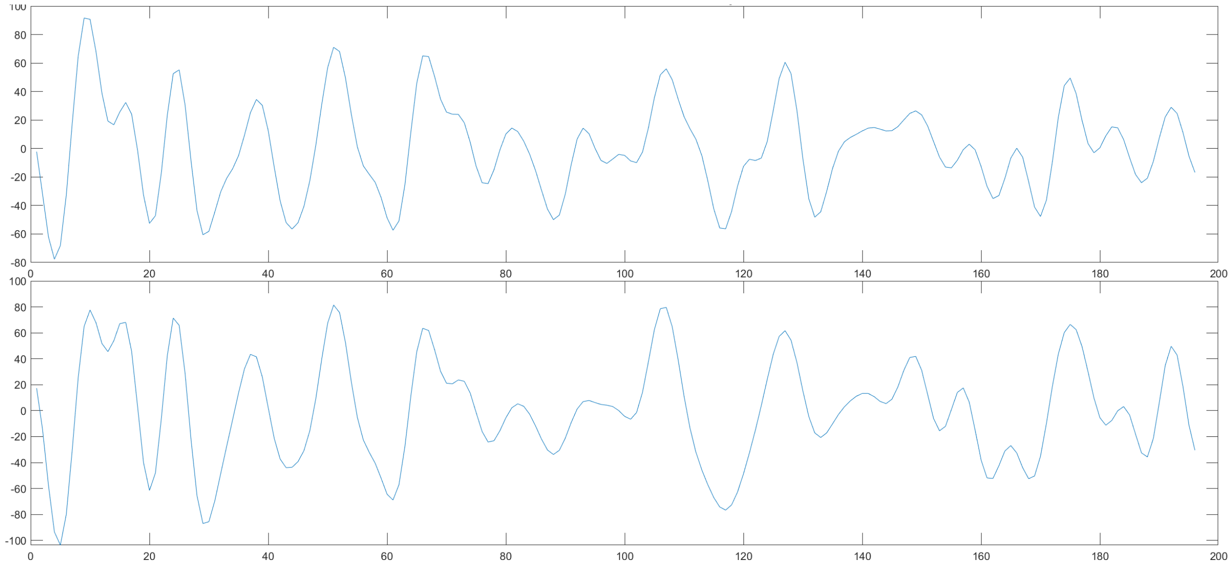


Figure 4.1. BOLD time series of highest correlated regions.

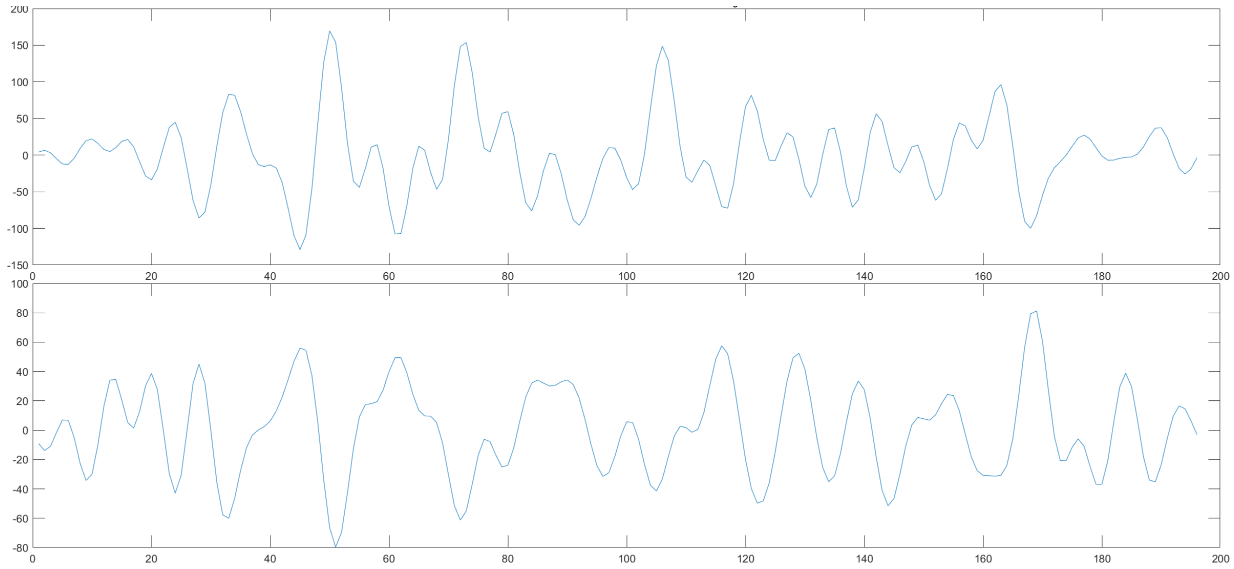


Figure 4.2. BOLD time series of least correlated regions.

CHAPTER 5

METHODOLOGIES

The architecture implemented is a 2-stage architecture.

i). CNN Ensemble Component architecture ii). 3D-CNN Temporal Convolutional Network (3D-CNN TCN)

5.1 Data Used for Component CNN Architecture

The fMRI time-series length for subjects across different imaging sites varies. To address the data shortage problem, the time-series length (tsl) of each subject was sliced with a constant window length (ws) and constant skip which produces floor((tsl – ws)/skip) + 1 number of data points per subject. The values of the window length and skip are 26 and 13 respectively which extracts a total of approx. 10000 datapoints across 10 folds. It is important to note that the data augmentation should be performed after segregating the original data into folds (in case of using k-fold cross validation strategy) so as to maintain the significance of each datapoint from the original data distribution. Further correlations are calculated to obtain functional connectivity matrix of dimensions 392x392.

5.2 CNN Ensemble Component Architecture

The Stacked CNN architecture was inspired by [20] where they implemented 7 CNN layers together stacked with each of the 7 layers having 400 filters of dimensions 1x392 to 7x392. The CNNs were followed by max-pooling and dropout layers with only 25% of the nodes active while training to reduce the number of features and avoid overfitting. The unique kernel-sizes, n x 392, used for each CNN, where n represents nth CNN from the 7 CNNs, represents the connections of n regions near each other with other regions. For our proposed architecture we picked 3 CNN layers out of these 7, namely the 1st, 4th, and 7th CNN layer with 3 kernels of the same respective kernel sizes. These 3 CNNs are followed by max-pooling layers of kernel sizes (392x1), (389x1) and (386x1), respectively. The 392x392 input matrix, after each CNN and max-pooling layer, is reduced to features of dimension 1x1 for each kernel. The output of these 3 CNNs are concatenated, flattened, fed to a Dense layer with 3 units and then finally to a sigmoid layer for classification. A Dropout layer reducing 30% of the units is added before the Dense layer. Hyperbolic tangent activation is used to preserve the information about functional connectivity after CNN and Dense layers. Batch size of 150 and learning rate of 0.0009 are employed to stabilize validation loss fluctuations during the network's pre-training over 60 epochs. To address the overfitting problem and to learn robust features, a noise layer with 0 mean and 0.9 standard deviation is proposed with different random seed for every input batch. The model is selected based on the following criteria: i). The model has approximately 60% accuracy on both training (noise free) and validation set indicating sufficient feature learning. ii). The model has not overfitted or has not captured noise from the training data.

The component CNN architecture is shown in Figure 5.1.

5.3 Data Used for 3D-CNN TCN Architecture

In order to investigate the temporal aspect of the rs-fMRI data, we suggest the following data preprocessing and architecture.

i). Data length standardization: A constant time-series length is required for all the data points, therefore a time-series length should be long enough for analy-

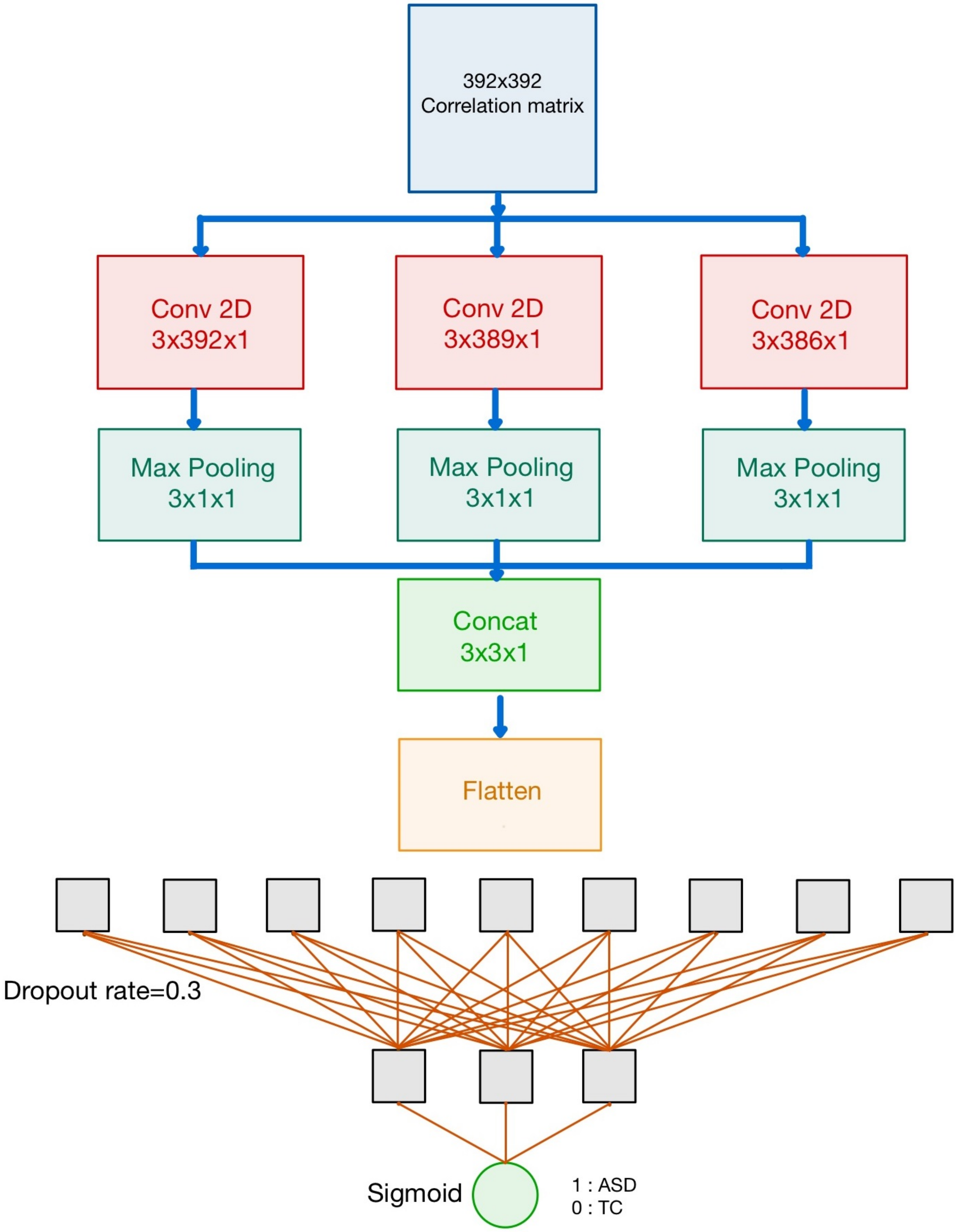


Figure 5.1. Component CNN Architecture.

sis and short enough for segmentation across most of the sites. We are selecting time-series of length 116 for segmentation across all the sites. Note that datapoints with time-series length shorter than 116 will be lost. Further, to avoid loss of data from longer time-series, we perform augmentation through window segmentation as described in the previous section. Here we use 116 as the window size and a skip size of 50.

ii). Creating temporal depth for functional connectivity analysis over time: Similar to data augmentation using window segmentation, we are using windows of size 26 with a skip of 6, producing a depth of 16 for a data point with time-series length of 116. The 16 segments obtained are stacked in the temporal dimension forming datapoints of dimension (26x392x16). Here 26 is the length of each element in the time-series, 392 are the number of ROIs and 16 is the temporal depth. Now, for every temporal point we calculate the connectivity matrix using PCC resulting in datapoints with dimension (392x392x16). This functional connectivity sequence is used as input to the temporal convolutional network for examining the effect of the temporal dimension on classification of ASD patients.

5.4 3D-CNN Temporal Convolutional Network (3D-CNN TCN)

A Temporal Convolutional Network is designed for sequence modeling tasks. Our temporal convolutional architecture is designed to process time series at a latent feature level, with the Component network constructing the latent features. Therefore, this network is further divided into 2 stages where the first stage uses a 3D-CNN that is a pre-trained component network for extracting the features from the correlation matrices stacked in the temporal dimension and the second stage is the actual temporal convolutional network which is used for classification. By expanding the dimension of each layer of the pre-trained component network and incorporating a temporal dimension we have produced a univariate sequence output of length 16. The Conv2d and max-pooling2d layers are upgraded to Conv3d and max-pooling3d, respectively, expanding the dimensions of the kernels to derive features from the depth dimension. The kernel sizes of convolution and max-pooling layers are selected to ensure that elements within the time dimension are processed independently. This ensures that kernels don't mix information from different time indices when processing the current time step. Figure 5.2 shows the architecture of this 3D-CNN architecture. The dense layers are stacked in a manner which also ensures that the temporal depth is operated on and classified independently. After pre-training the component network, the weights of each layer are expanded in dimension and frozen to retain the features captured. Further, the output of the 3D-CNN is passed to the TCN architecture. The TCN architecture consists of 5 Conv1D layers with a convolutional kernel of size 4. The 5 layers have 8, 16, 16, 32 and 32 kernels, respectively. The output of the last layer, which has 1 convolutional unit, is followed by a fully connected layer and a sigmoid layer. Figure 5.3 shows the architecture of the Temporal Convolution Component of the overall network. The training process is carried out for a total of 150 epochs. Each unit in the subsequent layer of the TCN has an increased visibility of its previous layer depending upon the size of the kernel used. Therefore, the last kernel has a visibility over the entire range of the input sequence.

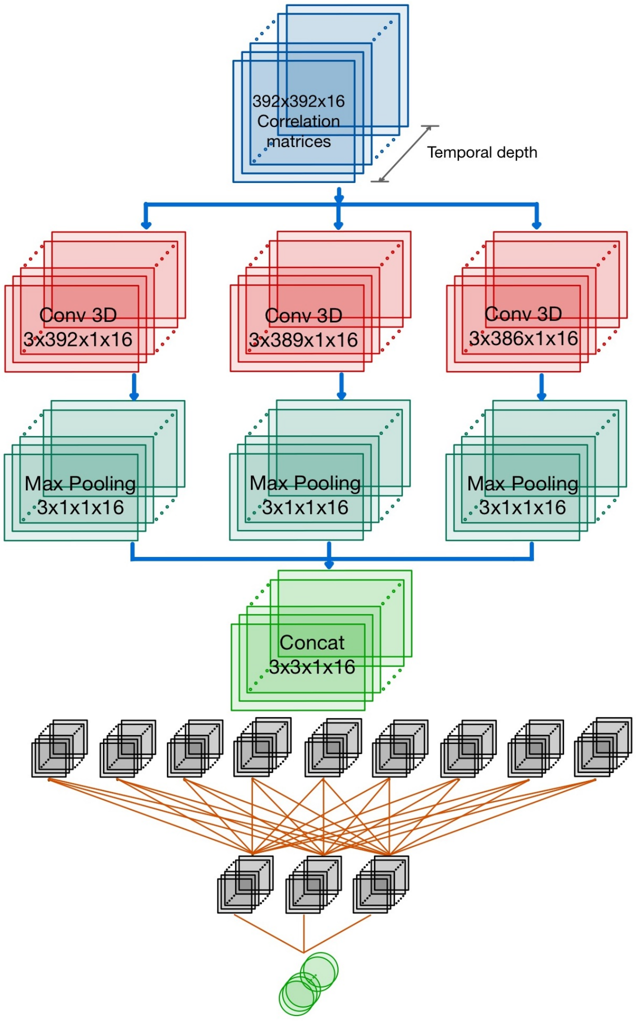


Figure 5.2. 3D-CNN Architecture.

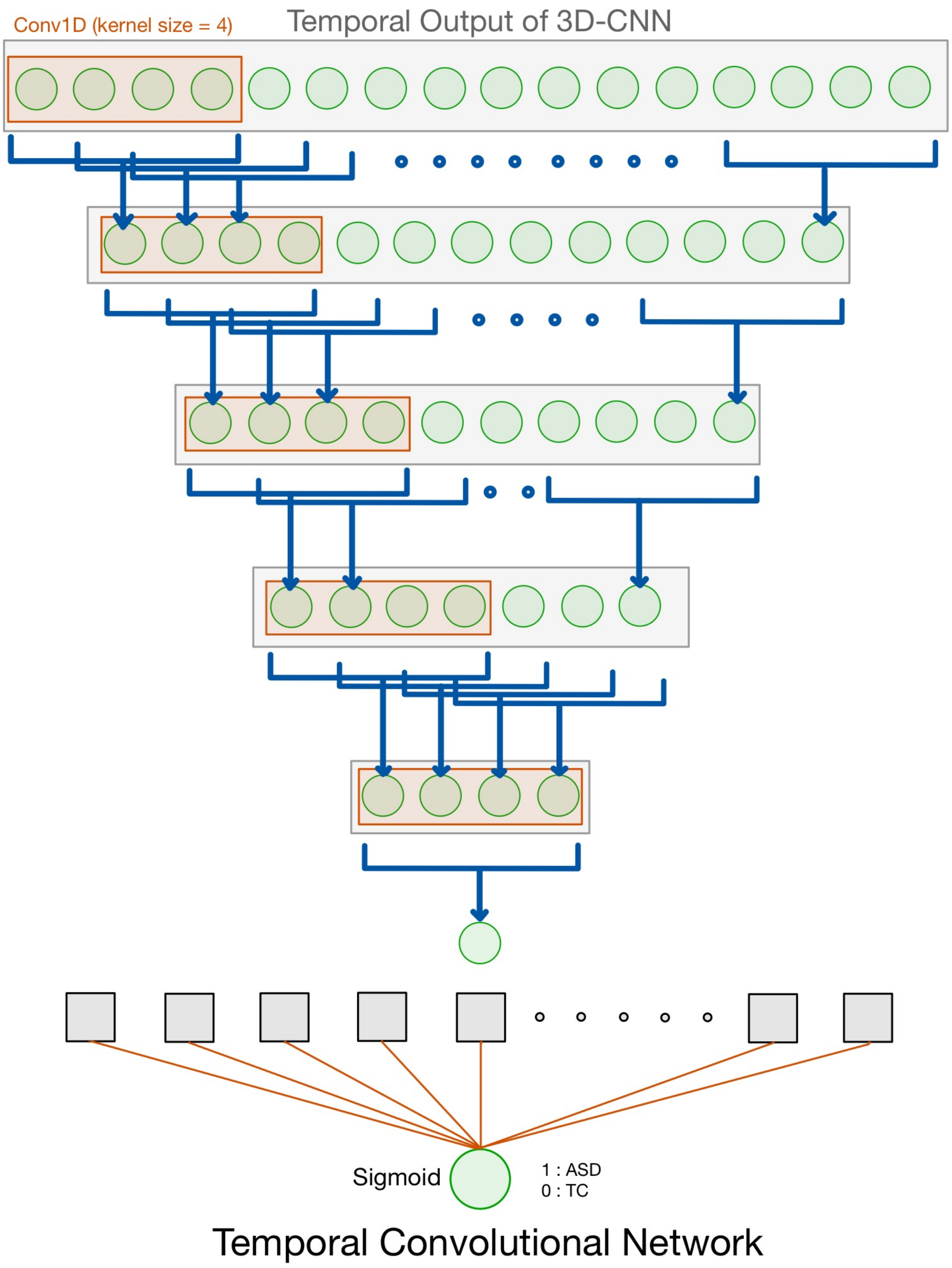


Figure 5.3. Temporal Convolutional Network.

CHAPTER 6

THE OVERFITTING PROBLEM

Due to the large size of the connectivity matrices that serve as an input for many of the rs-fMRI studies and the rather limited number of data points, most deep learning approaches that have been used for ASD prediction have suffered from significant overfitting which is usually addressed using early stopping and hand-selection of the final model. This potentially results in sub-optimal models and somewhat misleading classification accuracy numbers due to the high variance between different training runs. Reducing the amount of overfitting would therefore be useful and a number of methods have been tested and applied in this thesis.

6.1 Analysis of Existing Method

Our proposed CNN Ensemble Component architecture is mainly inspired by the CNN architecture presented by [1]. To analyze the network proposed by [1] (shown in Fig. 6.1) we implemented the same network which consisted of 7 stacked CNNs, each with connectome kernels ranging from size 1x392 to 7x392 for each CNN, respectively, and hyperparameters including 400 kernels, 0.005 learning rate, 32 batch size and 0.75 dropout rate.

We have incorporated an early stopping mechanism which terminated the training when the training accuracy reached 99%. The results of this network using 10-fold cross validation suggested overfitting in the network rendering the model to capture noise instead of extracting the underlying patterns representative of ASD. Fig 6.2 show the performance of the model on one of the folds, where folds were shuffled with

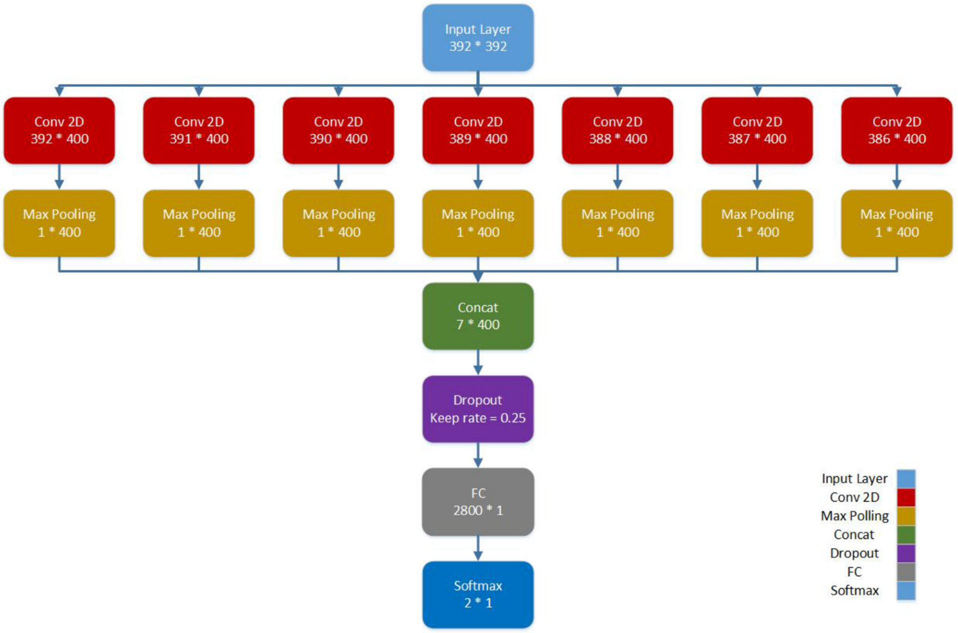


Figure 6.1. Full CNN Architecture from [1].

a random seed of 42. Due to the early stopping applied, the training lasted from 8 to 10 epochs for all the 10 folds.

6.2 Proposed Methods to Address Overfitting

In order to tackle the overfitting problem, we have introduced the below 2 approaches :

i). Data Augmentation through time-series segmentation with equal intervals as discussed previously in the Methodologies section.

ii). Noise Addition with different random seeds for every batch of every epoch.

6.2.1 Data Augmentation Through Time-Series Segmentation with Equal Intervals

Since the data is aggregated from multiple sites, the time-series length of the BOLD signal can differ in length. Segmentation is done for each of the 392

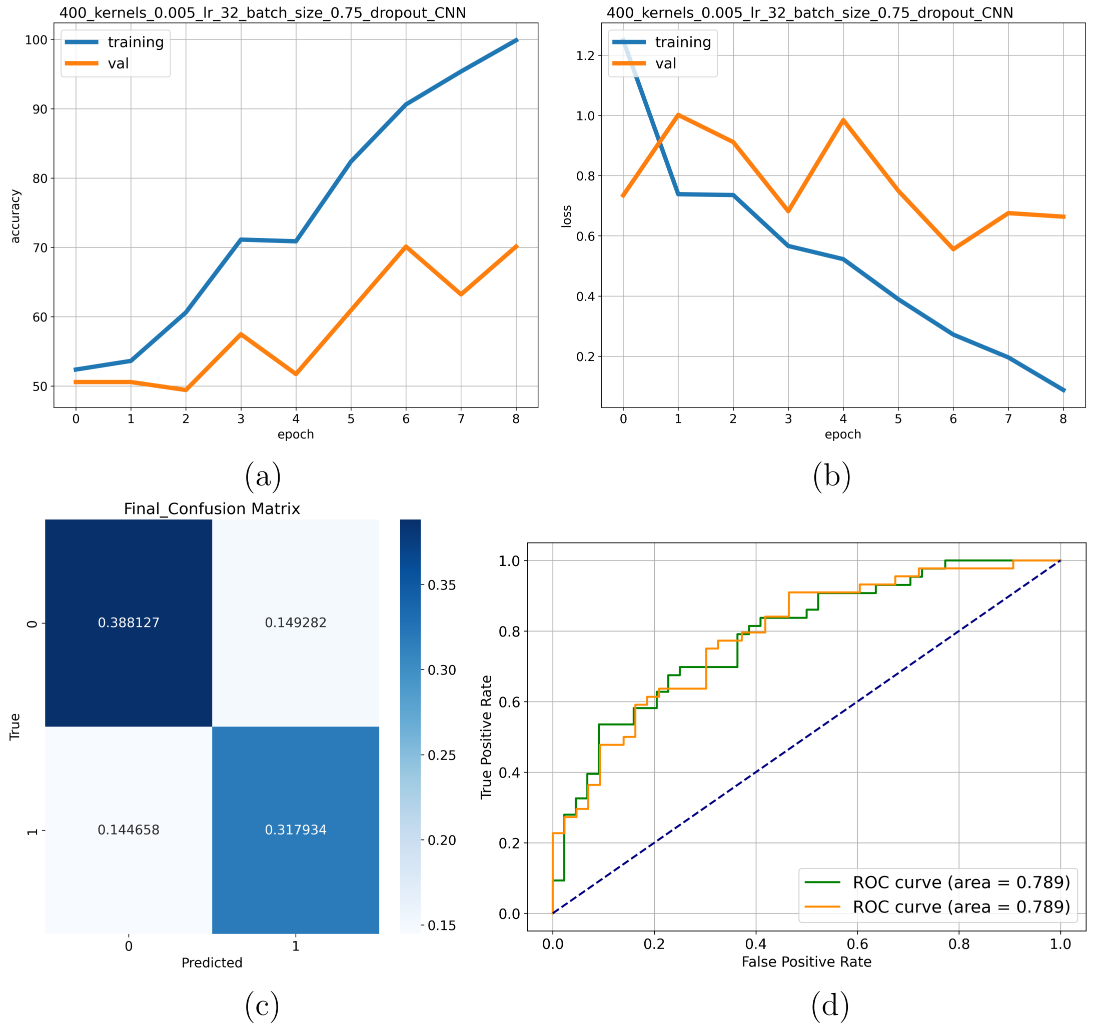


Figure 6.2. (a) Accuracy (b) Loss (c) Confusion Matrix (d) ROC curve for an example training run of the 3D-CNN architecture of [1].

brain regions and functional connectivity is calculated from each of the adjacent segments across the 392 brain regions of a subject. Therefore, based on the time-series length, the total number of segments that can be extracted from a subject is $\lfloor((timeSeriesLength–windowSize)/skip)\rfloor + 1$. Here we are using a window length of 26 and skip of 13 for each subject. Fig. 6.3 shows how time-series segmentation is performed on the BOLD signals obtained from one of the 392 brain regions of a subject. The window length is selected to be 26 to keep it the same size as the windows used to create the temporal dimension in the 3D-CNN TCN. The main dataset is first split into 10 folds and the functional connectivity matrices are obtained after augmentation through time-series segmentation. Since there are different time-series length of the brain ROIs of each subject, after data augmentation the distribution of the training set will be skewed towards the data of subjects which were captured from the sites with longer time-series length creating more data points. The splitting into 10 folds prior to data augmentation is necessary so as to maintain the distribution of the training and validation sets across 10 folds.

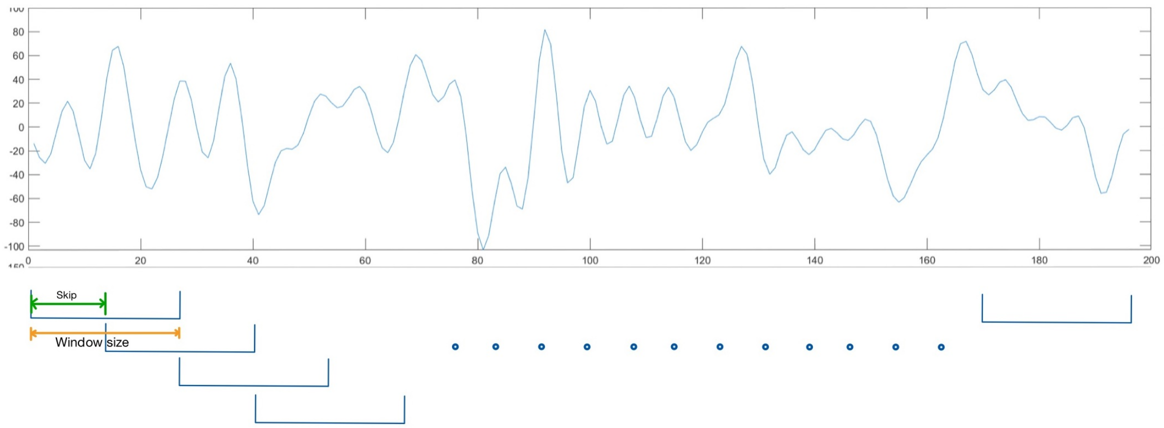


Figure 6.3. Time Series Segmentation for Data Augmentation.

6.2.2 Noise Addition

Random noise is added from a Gaussian distribution with 0 mean and 0.9 standard deviation. We have experimented with standard deviation in the range from 0.01 to 1.2 for injecting noise to augment the dataset and observed the value of 0.9 to have the required amount of effect on overfitting. A different random seed for each batch of each epoch is used to generate unlimited different data points so as to address the overfitting problem.

6.3 Component CNN Autoencoder for Regularization

Additionally, we experimented with an Autoencoder structure to induce regularization. Autoencoders are a type of neural network used to learn compressed and efficient representations of data for the purpose of dimensionality reduction, feature learning or data compression. The basic structure of an Autoencoder typically consists of an Encoder, a bottleneck or a latent representation, and a Decoder. An Encoder compresses the input data into a latent space representation. It defines mapping from input data to a latent space, referred to as bottleneck, while trying to preserve as much efficient information as possible. A Decoder on the other hand tries to reconstruct the original input from the bottleneck or latent space representation. It defines mapping from latent space to original input space [37], [38], [39]. Our initial CNN Ensemble Component network is used as a feature extractor for the main classification network and also as an encoder. Fig 6.4 shows the setup of our Encoder-Decoder classification architecture.

The last fully connected layer serves a different purpose than the bottleneck layer, functioning as a dense layer within the classification CNN architecture. This layer is used as input for the reconstruction phase of the Decoder. The process of unflattening involves reshaping the flat output from the last dense layer to match the dimensions of the Concatenate layer's output. Subsequently, this reshaped output is split along the same dimension used during concatenation in order to feed into the three parallel deconvolutional networks. To reverse the effects of max pooling, upsampling is applied using the same kernel size and 'nearest' interpolation method, which duplicates the nearest pixel's value from the input tensor. The 3 ConvTranspose2D layers act as deconvolution layers and recreate the correlation matrices which are then averaged out to reproduce the input functional connectivity matrix. The features learned by the encoder, which ideally represent the data well, can be fed into the classification network to improve its performance. The loss function used for this network is a weighted combination of Binary cross entropy loss (classification loss) and Mean Squared Error loss (reconstruction loss) shown below :-

$$\text{Loss} = \alpha \times \text{BCEloss} + \beta \times \text{MSEloss}$$

The total loss function ensures that the encoder learns features that are good for reconstruction and useful for the classification task. By forcing the network to learn a compressed representation of the data, autoencoders can help the main classification network to focus on the more important features. This implicitly performs feature selection, which can be a form of regularization, preventing overfitting to the noise in the data. Initially, to verify whether the Encoder-Decoder Architecture can learn significant features to reconstruct the original connectivity matrix, we discarded the classification loss by giving it 0 weight and experimented with different values of kernels to see if increasing the feature space can increase the chances of minimizing the reconstruction loss. We used the following values of kernel counts: 3 (same as in the proposed component architecture), 32, 128 and 400 ([1]).

Figure 6.5 shows the resulting learning curves for the different kernel counts. These figures show that the autoencoder does not achieve very good reconstructions. This is likely due to the sizes of the convolution kernels which stretch the entire length of the data and the use of large pooling windows in the component network architecture. This indicates that for this particular architecture this regularization technique might not contribute significantly. However, it might be a useful tool for other component networks in the future.

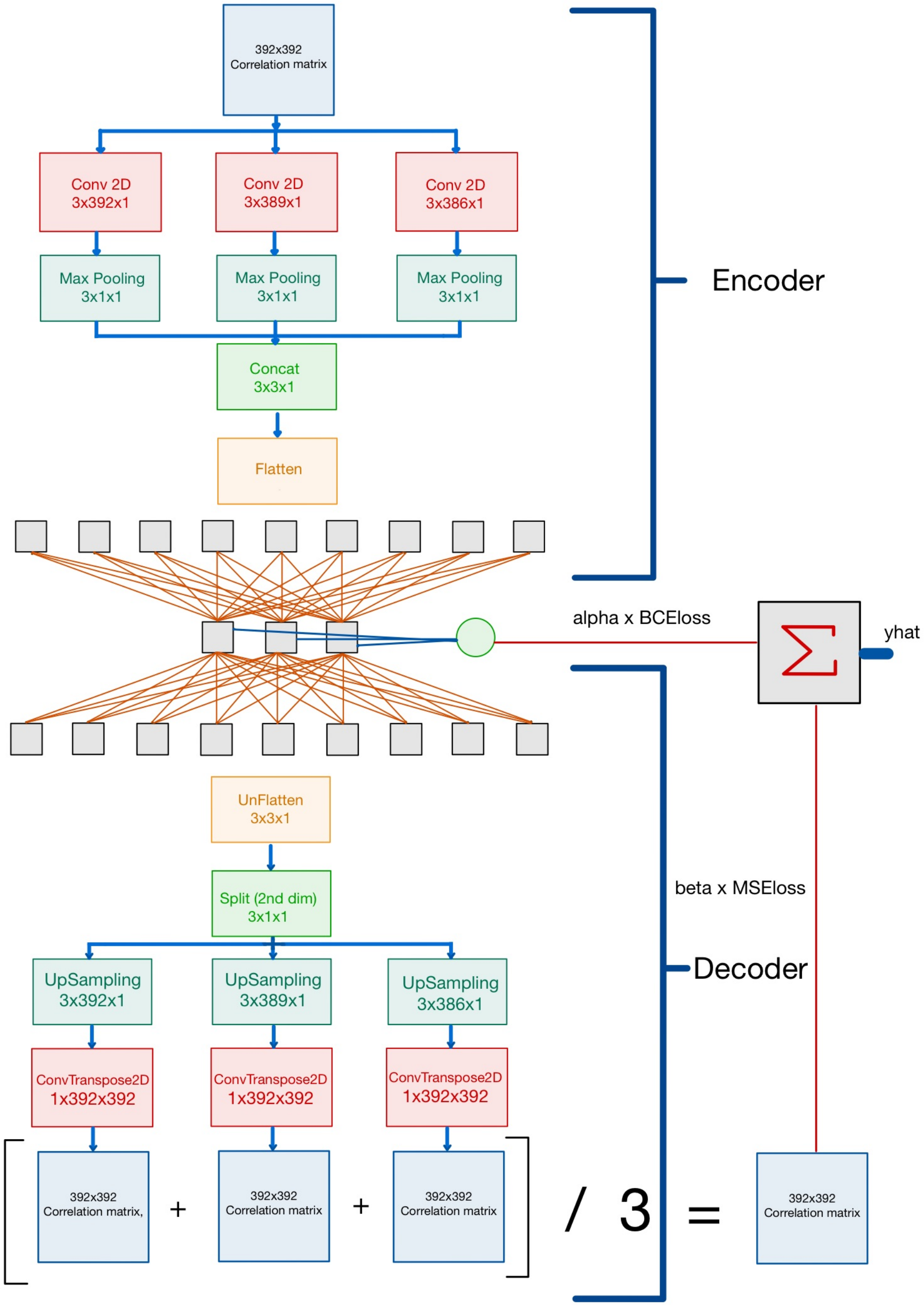


Figure 6.4. CNN Encoder-Decoder Architecture.

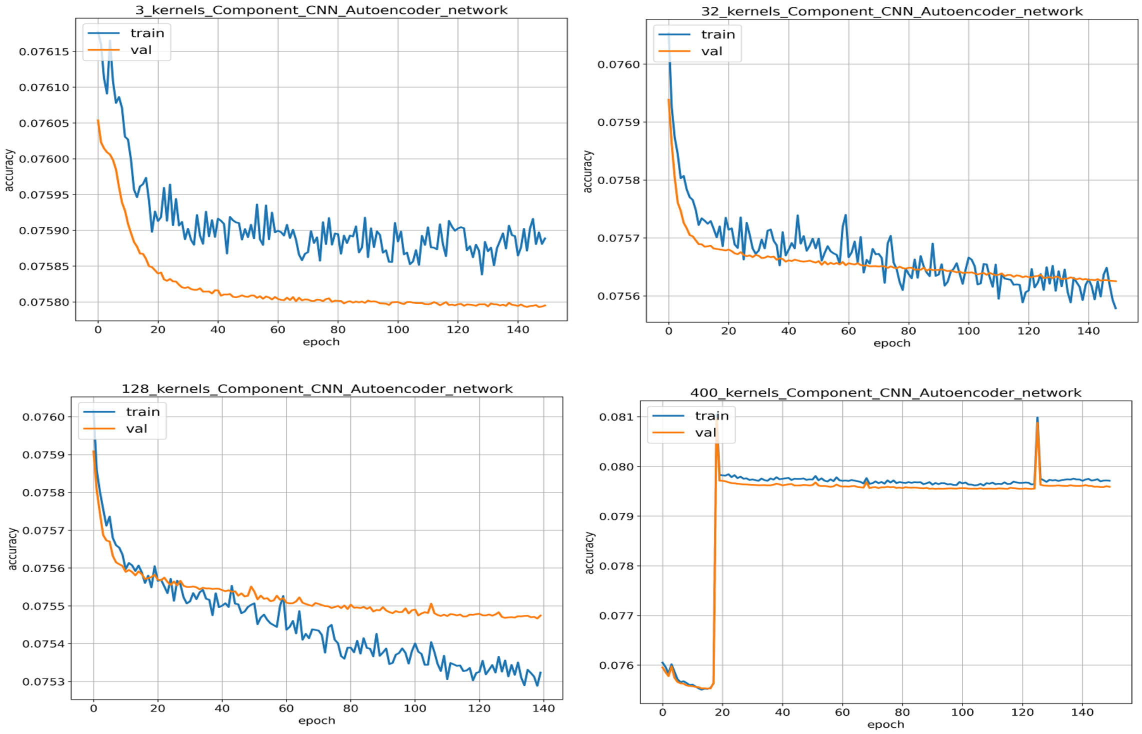


Figure 6.5. Reconstruction Losses on Different Kernel Counts of 3, 32, 128, and 400 (From Top Left to Bottom Right).

CHAPTER 7

RESULTS

In this study we implemented a 2-stage network and trained it by i). pre-training an Ensemble CNN network for extracting features, and ii). freezing the pre-trained network weights and applying it as a convolution kernel on the temporal dimension by converting the 1D layers to 2D and, 2D layers to 3D for an Ensemble 3D-CNN network. The output produced by this network will be similar to that of the component ensemble CNN network with an additional temporal dimension incorporated into it. Further, as part of the Temporal Convolutional Network, we applied a Conv1D layer on the temporal dimension which resulted from the 3D-CNN with a kernel size of 4. We applied a 10-fold cross validation strategy to evaluate and compare our model with other studies. To perform 10-fold cross validation on a 2-stage network we used the following approach:

i). Model selection is based on overall accuracy on the training and validation set. Note: The training accuracy is not used for model selection since the model is learning on data induced with noise. After each epoch the model is evaluated again on the training and validation set without noise and these metrics are used for model selection. For each of the 10 folds in the performed 10-fold cross validation we handpick models by observing the training and overfitting trend.

ii). The models selected from the 10 folds after evaluating the data on the component network are used as part of the frozen 3D-CNN. It is important to make sure that the model selected from a particular fold of stage 1, should only be used on the same fold during stage 2 of our architecture in order to prevent data leakage and misevaluation of the architecture. The results of the 10-fold cross validation obtained after performing temporal convolution are used to evaluate our complete architecture.

7.1 Architecture Stage 1 Results

Fig. 7.1 shows the results of the component Ensemble CNN network for each of the 10 folds and the epochs on which the models were selected (circled in red). Table 7.1 shows the precise accuracy for the 10folds at which the models were selected :

Fold	**Training Accuracy**	**Validation Accuracy**
1	61.04%	59.38%
2	64.26%	64.82%
3	62.88%	60.60%
4	61.55%	57.54%
5	62.65%	59.90%
6	64.24%	64.41%
7	64.57%	64.24%
8	65.34%	65.88%
9	59.86%	58.01%
10	64.66%	62.01%

Table 7.1. Model Selection Accuracy for 10-fold Cross Validation

From the graphs in Fig. 7.1 and the model selection criteria shown in Table 7.1, it can be observed that the model performs differently across folds. This is mainly due to shortage and inter-site variability of the data. This has resulted in difficulty for a model to perform uniformly across 10 folds. Our aim at this point is to create a model which has minimal overfitting and is capable of extracting features to analyze temporal patterns across the time-series using our proposed Temporal Convolutional Network model. We are considering validation accuracy of approximately 60% to be a sufficient indication of feature extraction, and accuracy difference between noise-free

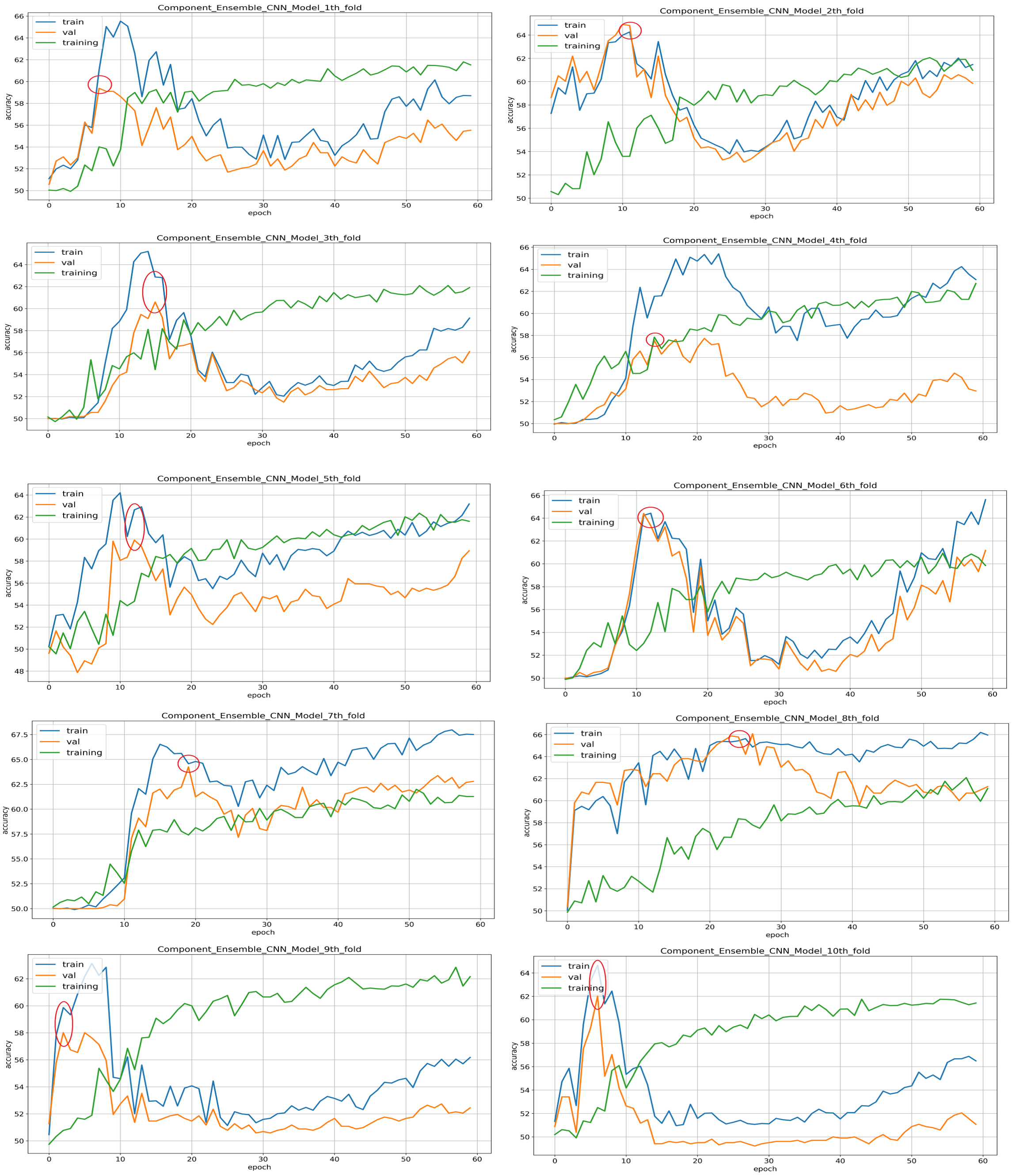


Figure 7.1. Train, Training, and Validation Accuracy Plots for All 10 Folds and Point of Model Selection, Indicated with Red Ellipse.

training set and validation set of +/- 4% to be acceptable with respect to overfitting. This is taken into consideration after our analysis on the model proposed by [1] where the difference between training and validation accuracy was approximately 20%, signifying overfitting, as shown in the Fig. 6.2. Figure 7.2 shows more detail of the performance of the selected component models in the form of the confusion matrices of the selected models for all 10 folds.

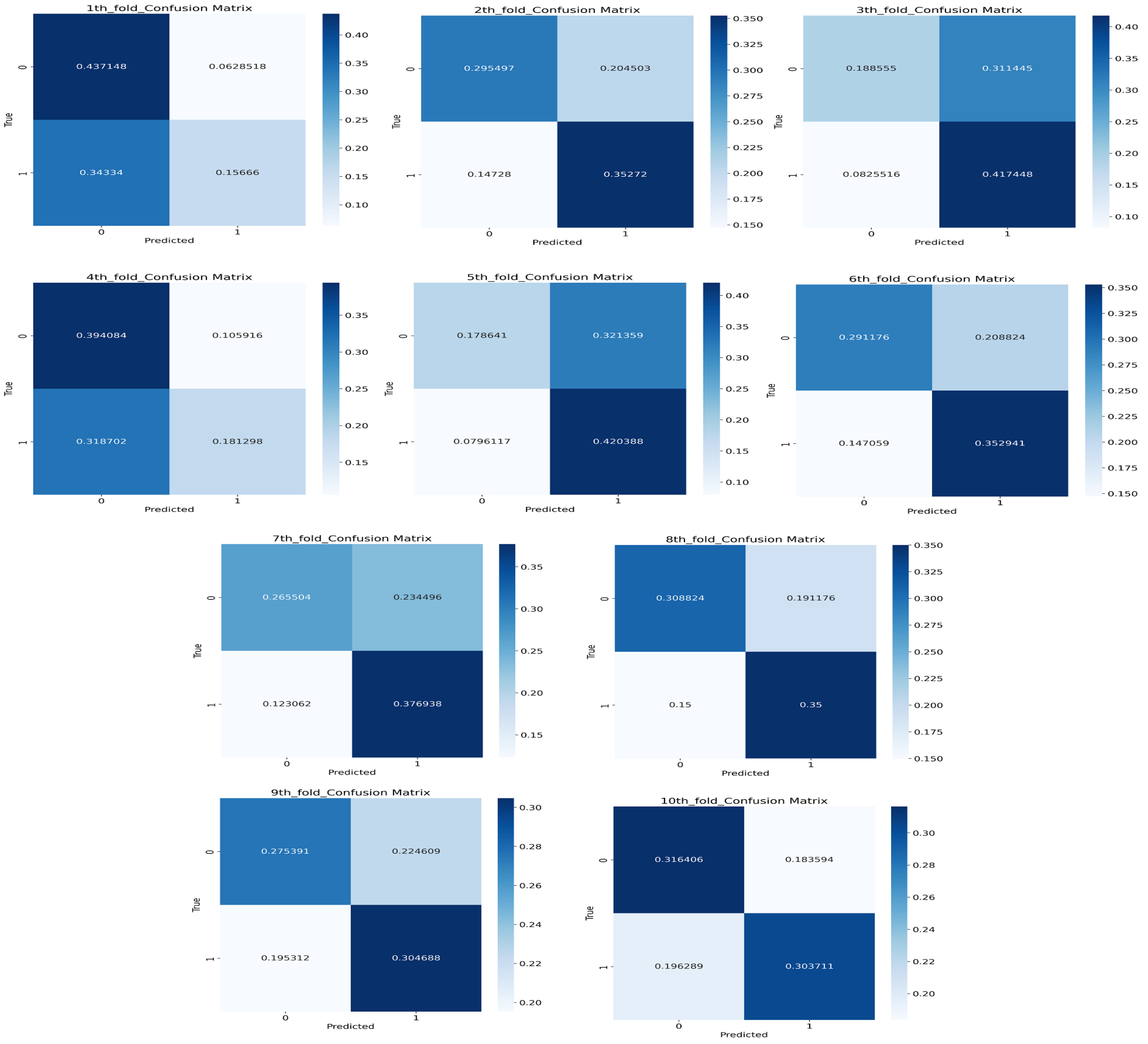


Figure 7.2. Confusion Matrices of Ensemble Component CNN (10 folds).

Again the results show significant variance across the 10 folds. This variance caused by the aggregation of inter-site data for our dataset can also have an effect on our temporal model which will be discussed further.

7.2 Architecture Stage 2 Results

Using the component models selected as above, their weights were frozen and they were integrated into the Stage 2 TCN network. The resulting performance of the 3D-CNN + TCN models during training is shown in Figure 7.3. Final performance for different metrics for each of the 10 folds is shown in Table 7.2.

Folds	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**	**F1 Score**	**ROC-AUC**
1	67.44%	67.05%	68.60%	66.28%	67.82%	0.67915
2	71.51%	72.84%	68.60%	74.42%	70.66%	0.76257
3	65.12%	63.83%	69.77%	60.47%	66.67%	0.68361
4	64.88%	62.63%	73.81%	55.95%	67.76%	0.66015
5	66.46%	64.84%	71.95%	60.98%	68.21%	0.62731
6	69.14%	66.67%	76.54%	61.73%	71.26%	0.72672
7	70.48%	71.79%	67.47%	73.49%	69.57%	0.75207
8	74.07%	74.07%	74.07%	74.07%	74.07%	0.77854
9	64.63%	63.64%	68.29%	60.98%	65.88%	0.67728
10	65.24%	65.82%	63.41%	67.07%	64.06%	0.64396
Average	**67.90%**	**67.07%**	**70.25%**	**64.90%**	**68.60%**	**0.7036**

Table 7.2. 10-Fold Cross-Validation Results (3D-CNN TCN)

Analysis of this performance data for each fold yields:

Fold 1: Both training and validation accuracy shows high variability initially and seems to converge over time with slight fluctuations. The difference between training and validation accuracy during model selection after pre-training is 1.66% which indicates minor overfitting. This slight overfitting is reflected in gains and per-

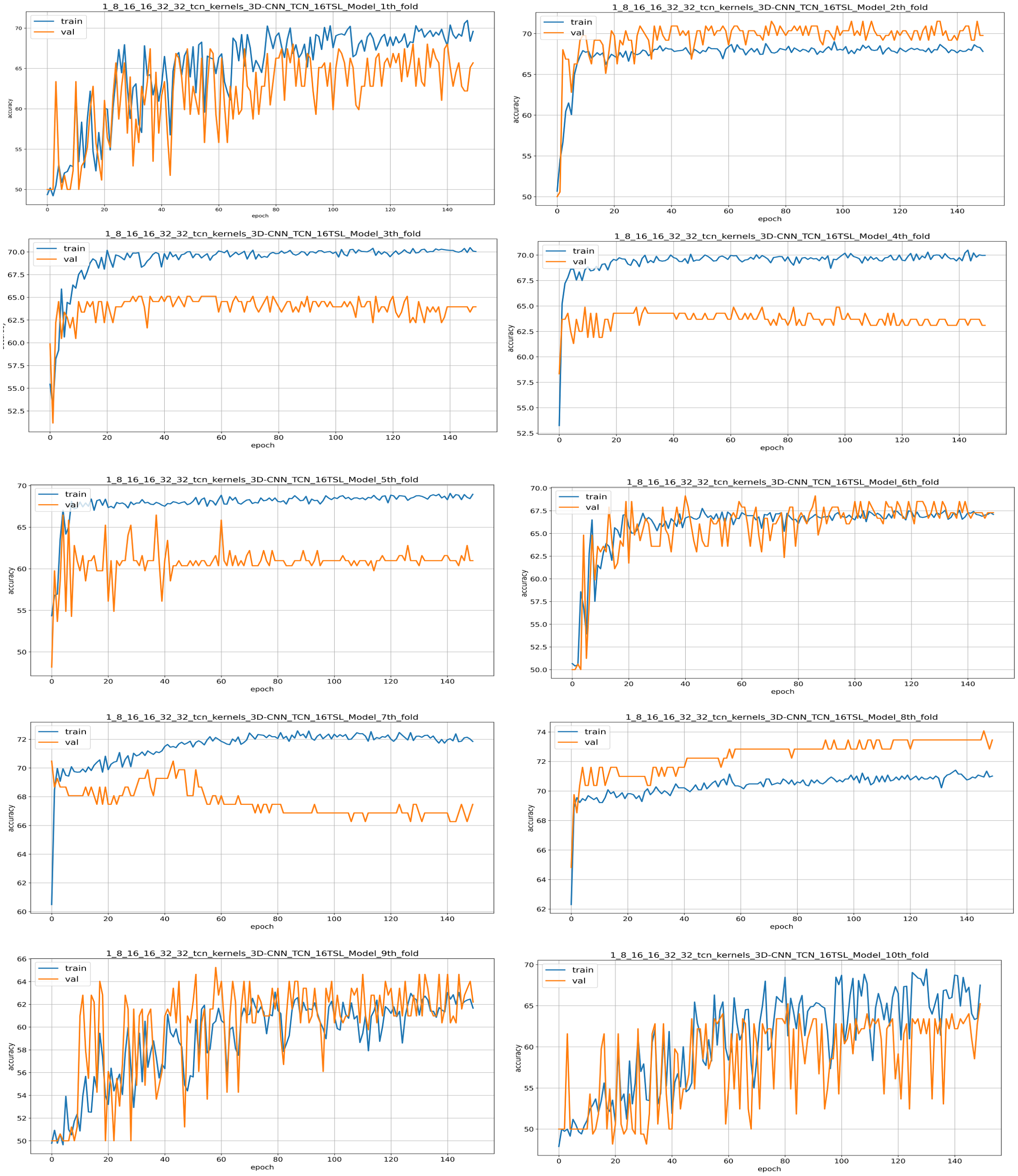


Figure 7.3. Accuracy Plots (TCN Architecture) for All 10 Folds.

formance of the TCN model. The model selection on the TCN model is done when the model stabilized towards the end of training.

Fold 2: During pre-training the model is selected at a validation accuracy slight higher than the training accuracy. Same is reflected in the TCN's gain. The model converges early, plateaus and is stable till the end of training.

Fold 3: Pre-training model selection has a difference of 2.28% in train-validation accuracy which is leading to the model producing higher gain for the training set and stabilizing. It can be observed that the TCN is capturing features through the temporal depth and able to produce a gain of approx. 5%.

Fold 4 and 5: Similar to Fold 3, since the Stage 1 Pre-training model selection has a difference of 4.01% and 2.75% respectively, in train-validation accuracy, the leading to the apparent difference between the training-validation accuracy and stabilizing overtime. It can be observed that the TCN is capturing features through the temporal depth and able to produce a gain of approx. 5%.

Fold 6: This fold has a very minor difference in training and validation accuracy of 0.17% indicating the least overfitting. Therefore it is reflecting similar performance with the gain in accuracy produced by the TCN.

Fold 7: The model selection was performed after a dip in training accuracy to prevent picking a pre-trained model with noise. This has seemed to influence the performance of the TCN where it exhibits a slight fall and then stabilization in accuracy over time.

Fold 8: Performance is similar to Fold 2 where the model selection is performed with validation accuracy 0.54% higher than training accuracy resulting in the same behaviour in TCN output.

Fold 9 and 10: Performance is similar to Fold 1 where the the Stage 1 model's training accuracy converges and overfits faster resulting in high variability with gain produced by the TCN.

To further study the performance characteristics of the resulting models, again confusion matrices were computed and compared in Figure 7.4.

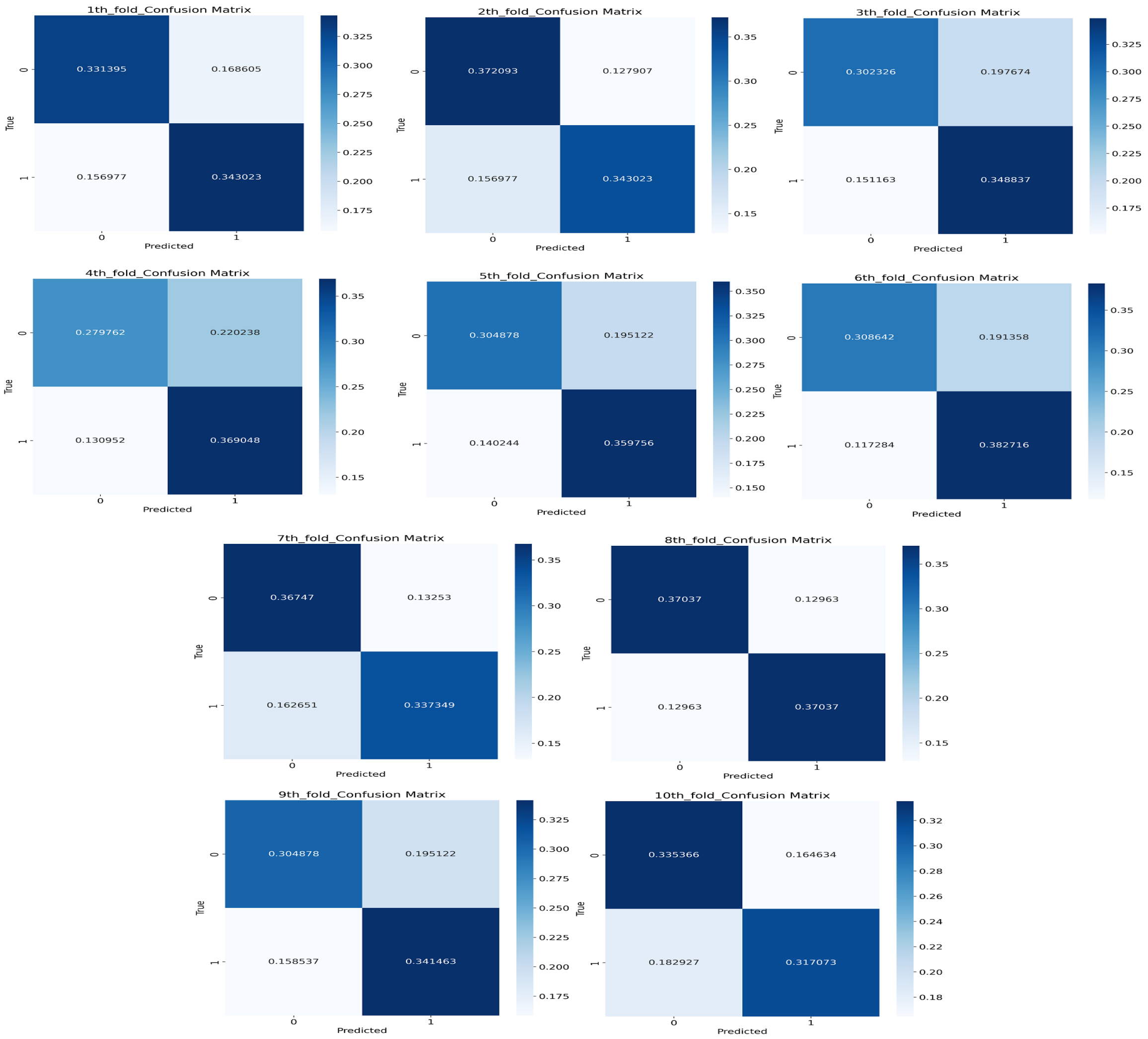


Figure 7.4. Confusion Matrices of TCN Architecture (10 Folds).

A Confusion Matrix is a tabular representation of the performance of a classification algorithm. It compares the actual target values against the values predicted by the model, providing insight into the types of errors produced by the classifier. The four quadrants of the matrix are as follows:

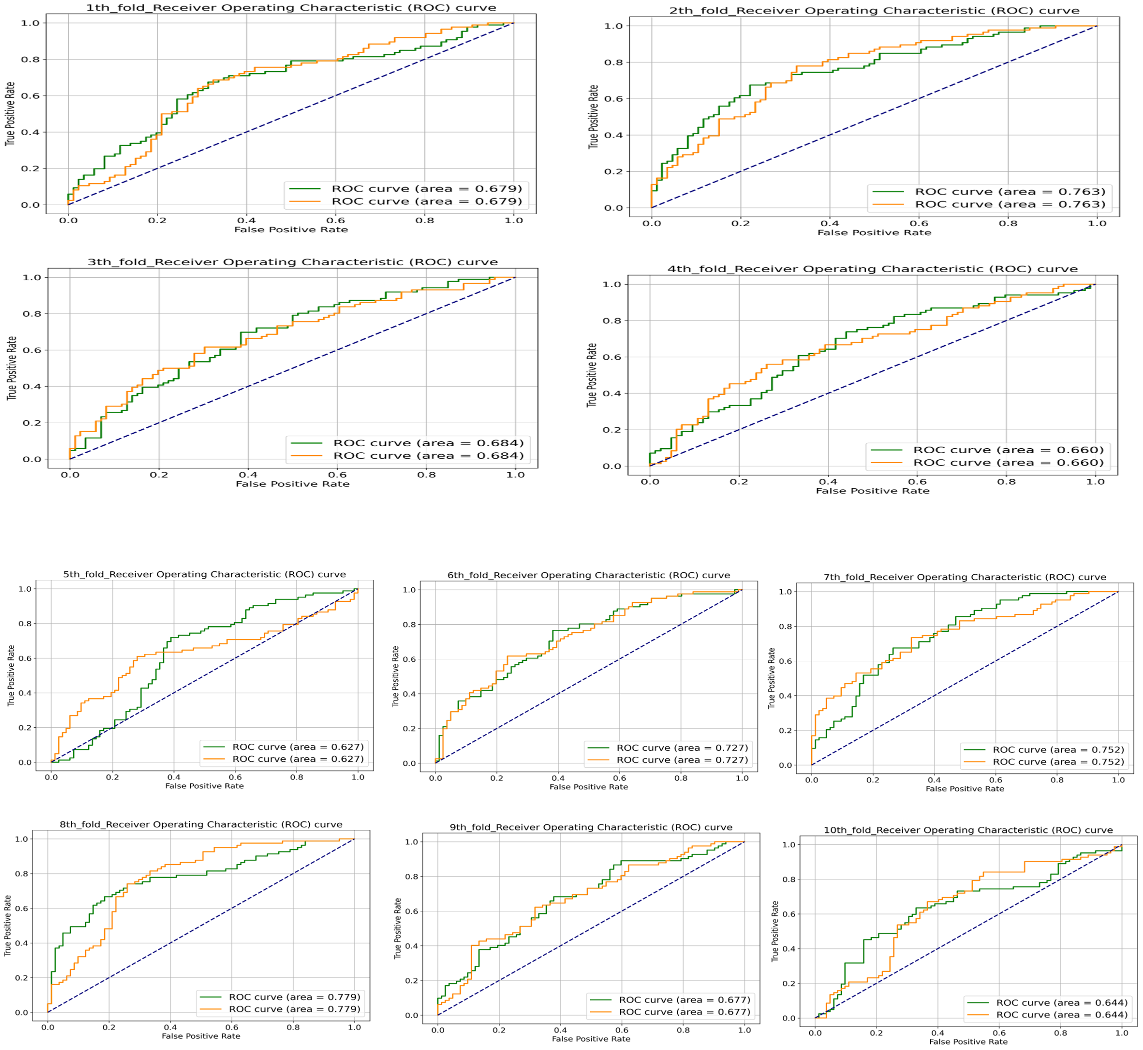
1. True Positives (TP): Observations correctly predicted as positive.
2. True Negatives (TN): Observations correctly predicted as negative.
3. False Positives (FP): Observations incorrectly predicted as positive.
4. False Negatives (FN): Observations incorrectly predicted as negative.

Various metrics such as Accuracy, Precision, Sensitivity (Recall), Specificity and F1-score can be calculated which provides unbiased view of the model's performance. Visualizing the heat-map can further help in interpreting these results, with color intensities reflecting the magnitude of values in the matrix, allowing fine-grained assessment of the model's capabilities. The diagonal (TP and TN) represent the correct predictions, while the other diagonal (FP and FN) indicate the incorrect predictions.

Comparing the results for the 10 foldsof Stage 1 (Ensemble Component CNN) shown in Fig. 7.2 and the results for the 10 folds of Stage 2 (3D-CNN TCN) shown in Fig. 7.4, it can be seen that the TCN model has improved the performance of the Stage 1 (feature detector CNN) model in terms of specificity and sensitivity as well.

Another way to see the training and final performance is by examining Receiver Operating Characteristics shown in Figure 7.5.

Receiver Operating Characteristics curve are a graphical representation of the diagnostic ability of a binary classifier as its discrimination threshold is varied. The curve represents the performance of the model by plotting the True Positive rates, also known as sensitivity, against the False positive rates , known as 1 - Specificity, for threshold values ranging from 0 to 1. True positive rates are basically the correctly classified positive class over total data points in the positive class and is calculated as the ratio of True Positives and the sum of True Positives and False Negatives. In contrast, False Positive rates is basically the incorrectly classified positive class over


1th_fold_Receiver Operating Characteristic (ROC) curve

2th_fold_Receiver Operating Characteristic (ROC) curve

3th_fold_Receiver Operating Characteristic (ROC) curve

4th_fold_Receiver Operating Characteristic (ROC) curve

5th_fold_Receiver Operating Characteristic (ROC) curve

6th_fold_Receiver Operating Characteristic (ROC) curve

7th_fold_Receiver Operating Characteristic (ROC) curve

8th_fold_Receiver Operating Characteristic (ROC) curve

9th_fold_Receiver Operating Characteristic (ROC) curve

10th_fold_Receiver Operating Characteristic (ROC) curve

Figure 7.5. Receiver Operating Characteristics of TCN architecture (10 Folds).

total negative data points and is calculated as the ratio of False Positive and the sum of False Positives and True Negatives [40].

The ROC curve provides us with the understanding of operational attributes of a model and helps in finding the threshold that's the most suitable for the operating requirements. This is particularly useful when the importance of false positives and false negatives varies. Falsely classifying a patient as typical control can be more expensive than falsely classifying a patient as positive for ASD.

Area under the ROC curve (AUC) can give an aggregate measure of performance across all possible classification thresholds. The AUC measures the entire two dimensional area from (0,0) to (1,1) under the ROC curve. An area on 0.5 represents no discrimination ability and is similar to random chance, whereas an area of 1 indicates a perfect classifier [41]. Practically, AUC represents the probability that the model ranks a random positive instance more highly than a random negative instance. Since it aggregates a range of thresholds, it is not influenced by the skewness of the dataset making it a robust measure for unbalance sets. Fig. 7.5 shows relatively robust performance of our model especially for the 2nd, 6th, 7th and 8th fold in terms of AUC.

7.3 Verification on Shuffled Temporal Dimension

To investigate if the enhancement in the classification accuracy is due to the temporal sequence information present in the time-series or a result of looking at the use of connectivity over shorter periods of time, we performed an experiment where the temporal dimension in the correlation matrices is shuffled. Therefore, the input with dimensions 392x392x16 will have its 392x392 matrices randomly arranged to examine the cause of its improvement in classification.

Results in Table 7.3 show that there are very minor differences in the results obtained with and without shuffling the temporal dimension of the input correlation matrix. This implies that the improved performance can be attributed to data available in shorter time frames, which gets averaged when computing the functional connectivity matrix over longer periods. Additionally, this finding suggests the potential for even more significant information in even shorter time windows, although caution is advised to avoid reducing the window length too much, as this could result

Fold	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC-AUC
1	68.02%	71.83%	59.30%	76.74%	64.97%	0.66888
2	74.42%	75.00%	73.26%	75.58%	74.12%	0.77528
3	65.70%	65.17%	67.44%	63.95%	66.29%	0.6824
4	64.29%	63.04%	69.05%	59.52%	65.91%	0.65476
5	64.02%	62.11%	71.95%	56.10%	66.67%	0.64396
6	65.43%	65.43%	65.43%	65.43%	65.43%	0.70294
7	68.07%	61.90%	93.98%	42.17%	74.64%	0.75018
8	73.46%	73.75%	72.84%	74.07%	73.29%	0.77884
9	65.85%	71.67%	52.44%	79.27%	60.56%	0.67876
10	67.07%	75.00%	51.22%	82.93%	60.87%	0.6749
Average	**67.63%**	**68.49%**	**67.69%**	**67.58%**	**67.28%**	**70.11%**

Table 7.3. 10-Fold Cross-Validation Results for Shuffled Sequence Order (3D-CNN TCN)

in the inclusion of noise into the calculation of the correlation matrices used as input to the classifier. Similarly, comparing the confusion matrices with the correct temporal sequence and randomized temporal sequence shown in Figure 7.6 shows only minor effects of the temporal sequence, again indicating that the gain in accuracy is more likely due to the use of shorter temporal windows than the temporal sequence order.

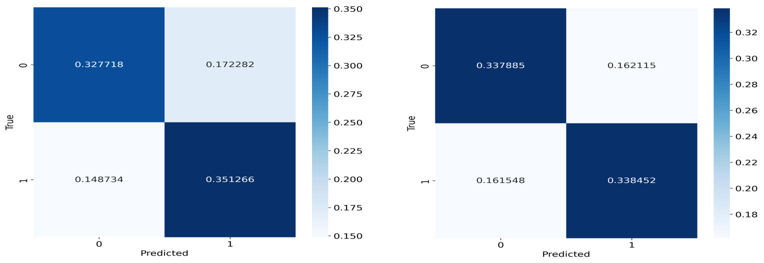


Figure 7.6. Confusion Matrix Comparison i). 3D-CNN TCN ii). 3D-CNN TCN with Shuffled Temporal Sequence Order.

7.4 Comparison Results

The performance and significance of the results obtained in this study are shown in table 7.4 and its corresponding confusion matrices in Fig. 7.7. We have compared the results obtained after Stage 1 (Component Ensemble CNN model) and Stage 2 (3D CNN-TCN model) along with the Component CNN model trained on correlation matrices of full length of the BOLD time-series, instead of segments, for 100 epochs.

Metrics/Methods	**Component CNN**	**Component CNN on full time-series**	**3D-CNN TCN**
Accuracy	61.68% (±2.86%)	63.83% (±2.95%)	67.90% (±3.10%)
Precision	62.08%	63.88%	67.07%
Sensitivity	64.34%	68.70%	70.25%
Specificity	59.03%	58.96%	64.90%
F1 Score	61.43%	64.58%	68.60%
ROC-AUC	0.6522	0.6618	0.7036

Table 7.4. Metrics Comparison Across i). Component Model ii). Component Model on Full Time-Series iii). 3D-CNN TCN Model

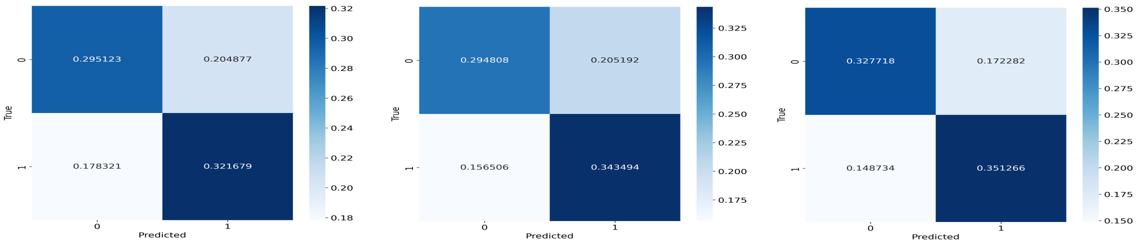


Figure 7.7. Confusion matrix Comparison across i). Component model ii). Component model on full time-series iii). 3D-CNN TCN model.

Investigating these results it can be observed that the results of the Component CNN model trained on full length time-series shows better performance than the

Component CNN (Stage 1) model trained on segments of the time-series, indicating the presence of either more relevant functional connectivity features in the full length time-series than in segments of the time-series or that only a subset of the segments in the Component CNN contain the markers for prediction. Looking at the confusion matrices, a small effect in the latter direction can be observed with higher false-positive and false-negative rates for the Component Network. To assess this further, comparison of the temporal convolution network and the component model with full time series information is performed. This shows that the performance of the 3D-CNN TCN model exhibits better performance than the Component CNN model trained on full length time-series, indicating that the features captured from the correlation matrices of time-series segments stacked together in temporal dimension are more indicative of ASD or TC than the features captured from the correlation matrix of the entire time-series. This indicates that there seem to be indicators that are present only for shorter segments and are averaged out when calculating the correlation matrix over the entire time series. The confusion matrices in Fig. 7.7 similarly show higher specificity and sensitivity of the 3D-CNN TCN model compared to the two cases of component models.

7.5 Comparison With Other Studies

Table 7.5 shows the comparison of our results across studies. We have mainly referred [1] for out study, fine tuned their model to reduce the overfitting, and used the same as our Component CNN (Stage 1) network. We have mainly focused here on investigating overfitting and the influence of temporal aspects on classification rather than raw prediction accuracy. The effect of this compared to our baseline comparison in [1] shows the former by producing a smaller spread between sensitivity and specificity. While the accuracy does not reach the value in some of the newer papers due to the efforts in avoiding overfitting, further tuning and selection of other component networks within the temporal convolution context might be able to increase overall performance above some of these baseline models while reducing the overfitting bias in the reported results.

Study	**Strategy**	**Accuracy**	**Sensitivity**	**Specificity**
Huang 2020 [27]	10 Fold CV	74.50% (± 2.2%)	76.20%	73.10%
Yang 2019 [32]	5 Fold CV	71.98% (N.A.)	70.89%	71.53%
Almuquim 2021 [31]	10 Fold CV	70.80% (N.A.)	62.20%	79.10%
Sherkatghanad 2020 [1]	10 Fold CV	70.22% (± 4.36%)	77.46%	61.82%
Eslami 2019 [28]	10 Fold CV	70.10% (N.A.)	67.80%	72.80%
Heinsfeld 2018 (DNN) [25]	10 Fold CV	70% (N.A.)	74%	63%
Proposed Method	10 Fold CV	**67.90% (±3.10%)**	**70.25%**	**67.07%**
Akhavan 2018 [29]	10 Fold CV	65.56% (N.A.)	84%	32.96%
Heinsfeld 2018 (SVM) [25]	10 Fold CV	65% (N.A.)	68%	62%
Nielsen 2013 [33]	LOOCV	60.00% (N.A.)	62.00%	58.00%

Table 7.5. Comparison Table of Studies for Detection of ASD and TC Using Functional Connectivity

CHAPTER 8

CONCLUSION AND FUTURE WORKS

8.1 Conclusion

In this study, we introduced a two-stage architecture designed to analyze temporal aspects within fMRI BOLD signals via functional connectivity. The pre-trained architecture is used for feature extraction and applying these features across the temporal dimension of the functional connectivity matrix, which possesses temporal depth. To assess the temporal patterns and enhance the classification performance, a temporal convolutional network was implemented. Using the ABIDE-I dataset, our model demonstrated an accuracy of 67.90%, with a specificity of 64.90% and a sensitivity of 70.25% over 10-fold cross-validation.

Furthermore, to determine whether the results stemmed from the temporal sequencing, we shuffled the temporal depth. Since the results obtained with and without shuffling the temporal depth had minimal difference, our experiments suggest a possibility of the patterns being more visible in shorter duration correlation matrices and loss of these patterns when computing the connectivity matrix on full length time-series. Furthermore, this also suggests the availability of patterns, distinguishing ASD patients from normal controls, in certain time intervals of the BOLD signals.

We also experimented with an encoder-decoder architecture of our component ensemble CNN to induce regularization. The experiments indicated that reconstructing the functional connectivity matrix from the latent representations of our architecture was challenging, suggesting that such reconstruction from connectome kernels may not be feasible. Finally, our approach of data augmentation through time-series segmentation at equal intervals and noise injection has shown to be conclusive in reducing overfitting of the model.

8.2 Future Work

Performance of the temporal convolutional network can be evaluated on a feature extractor model which is offering better feature extraction and more stability across fold by overcoming the shortage and inter-site variability of the data. The feature extractor considered, similar to our Stage-2 network, can be replicated across temporal dimension and the resulting features can be used on out proposed TCN.

Secondly, more investigation can be performed on the temporal depth to analyze if the ASD bio marker feature appears in specific intervals of the BOLD time-series.

Thirdly, the encoder-decoder architecture can be used to perform regularization and address the overfitting on other feature extractor models which have suitable layers and sufficient parameters for reconstruction.

REFERENCES

[1] Z. Sherkatghanad, M. Akhondzadeh, S. Salari, M. Zomorodi-Moghadam, M. Abdar, U. R. Acharya, R. Khosrowabadi, and V. Salari, "Automated detection of autism spectrum disorder using a convolutional neural network," *Frontiers in neuroscience*, vol. 13, p. 1325, 2020.

[2] C. Lord, S. Risi, P. S. DiLavore, C. Shulman, A. Thurm, and A. Pickles, "Autism from 2 to 9 years of age," *Archives of general psychiatry*, vol. 63, no. 6, pp. 694–701, 2006.

[3] R. E. Frye, S. Vassall, G. Kaur, C. Lewis, M. Karim, and D. Rossignol, "Emerging biomarkers in autism spectrum disorder: a systematic review," *Annals of translational medicine*, vol. 7, no. 23, 2019.

[4] M. Greicius, "Resting-state functional connectivity in neuropsychiatric disorders," *Current opinion in neurology*, vol. 21, no. 4, pp. 424–430, 2008.

[5] B. Biswal, F. Zerrin Yetkin, V. M. Haughton, and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar mri," *Magnetic resonance in medicine*, vol. 34, no. 4, pp. 537–541, 1995.

[6] R. J. Meszlényi, K. Buza, and Z. Vidnyánszky, "Resting state fmri functional connectivity-based classification using a convolutional neural network architecture," *Frontiers in neuroinformatics*, vol. 11, p. 61, 2017.

[7] M. I. Al-Hiyali, N. Yahya, I. Faye, and A. F. Hussein, "Identification of autism subtypes based on wavelet coherence of bold fmri signals using convolutional neural network," *Sensors*, vol. 21, no. 16, p. 5256, 2021.

[8] Y. Liang, B. Liu, and H. Zhang, "A convolutional neural network combined with prototype learning framework for brain functional network classification of autism spectrum disorder," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 2193–2202, 2021.

[9] G. Li, M. Liu, Q. Sun, D. Shen, and L. Wang, "Early diagnosis of autism disease by multi-channel cnns," in *Machine Learning in Medical Imaging: 9th International Workshop, MLMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 9.* Springer, 2018, pp. 303–309.

[10] J. Ji, Z. Chen, and C. Yang, "Convolutional neural network with sparse strategies to classify dynamic functional connectivity," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 3, pp. 1219–1228, 2021.

[11] A. Riaz, M. Asad, S. M. R. Al-Arif, E. Alonso, D. Dima, P. Corr, and G. Slabaugh, "Fcnet: a convolutional neural network for calculating functional connectivity from functional mri," in *Connectomics in NeuroImaging: First International Workshop, CNI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Proceedings 1.* Springer, 2017, pp. 70–78.

[12] K.-H. Oh, I.-S. Oh, U. Tsogt, J. Shen, W.-S. Kim, C. Liu, N.-I. Kang, K.-H. Lee, J. Sui, S.-W. Kim, *et al.*, "Diagnosis of schizophrenia with functional connectome data: a graph-based convolutional neural network approach," *BMC neuroscience*, vol. 23, no. 1, pp. 1–11, 2022.

[13] M. Bengs, N. Gessert, and A. Schlaefer, "4d spatio-temporal deep learning with 4d fmri data for autism spectrum disorder classification," *arXiv preprint arXiv:2004.10165*, 2020.

[14] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[15] K.-W. Park and S.-B. Cho, "A residual graph convolutional network with spatio-temporal features for autism classification from fmri brain images," *Applied Soft Computing*, vol. 142, p. 110363, 2023.

[16] R. M. Thomas, S. Gallo, L. Cerliani, P. Zhutovsky, A. El-Gazzar, and G. Van Wingen, "Classifying autism spectrum disorder using the temporal statistics of resting-state functional mri data with 3d convolutional neural networks," *Frontiers in psychiatry*, vol. 11, p. 440, 2020.

[17] S. Liu, S. Wang, H. Zhang, S.-H. Wang, J. Zhao, and J. Yan, "An asd classification based on a pseudo 4d resnet: Utilizing spatial and temporal convolution," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 9, no. 3, pp. 9–18, 2023.

[18] Y. Zhao, H. Dai, W. Zhang, F. Ge, and T. Liu, "Two-stage spatial temporal deep learning framework for functional brain network modeling," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019).* IEEE, 2019, pp. 1576–1580.

[19] S. Sarraf, D. D. DeSouza, J. Anderson, G. Tofighi, and A. D. N. Initiativ, "Deepad: Alzheimer's disease classification via deep convolutional neural networks using mri and fmri," *BioRxiv*, p. 070441, 2016.

[20] R. Ju, C. Hu, Q. Li, *et al.*, "Early diagnosis of alzheimer's disease based on resting-state brain networks and deep learning," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 1, pp. 244–257, 2017.

[21] P. Patel, P. Aggarwal, and A. Gupta, "Classification of schizophrenia versus normal subjects using deep learning," in *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, 2016, pp. 1–6.

[22] A. Boutet, R. Madhavan, G. J. Elias, S. E. Joel, R. Gramer, M. Ranjan, V. Paramanandam, D. Xu, J. Germann, A. Loh, *et al.*, "Predicting optimal deep brain stimulation parameters for parkinson's disease using functional mri and machine learning," *Nature communications*, vol. 12, no. 1, p. 3043, 2021.

[23] S. Koyamada, Y. Shikauchi, K. Nakae, M. Koyama, and S. Ishii, "Deep learning of fmri big data: a novel approach to subject-transfer decoding. 2015," *arXiv preprint arXiv:1502.00093*.

[24] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, *et al.*, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," *Molecular psychiatry*, vol. 19, no. 6, pp. 659–667, 2014.

[25] A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, and F. Meneguzzi, "Identification of autism spectrum disorder using deep learning and the abide dataset," *NeuroImage: Clinical*, vol. 17, pp. 16–23, 2018.

[26] N. C. Dvornek, P. Ventola, K. A. Pelphrey, and J. S. Duncan, "Identifying autism from resting-state fmri using long short-term memory networks," in *Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8*. Springer, 2017, pp. 362–370.

[27] Z.-A. Huang, Z. Zhu, C. H. Yau, and K. C. Tan, "Identifying autism spectrum disorder from resting-state fmri using deep belief network," *IEEE Transactions on neural networks and learning systems*, vol. 32, no. 7, pp. 2847–2861, 2020.

[28] T. Eslami, V. Mirjalili, A. Fong, A. R. Laird, and F. Saeed, "Asd-diagnet: a hybrid learning approach for detection of autism spectrum disorder using fmri data," *Frontiers in neuroinformatics*, vol. 13, p. 70, 2019.

[29] M. Akhavan Aghdam, A. Sharifi, and M. M. Pedram, “Combination of rs-fmri and smri data to discriminate autism spectrum disorders in young children using deep belief network,” *Journal of digital imaging*, vol. 31, pp. 895–903, 2018.

[30] X.-a. Bi, Y. Wang, Q. Shu, Q. Sun, and Q. Xu, “Classification of autism spectrum disorder using random support vector machine cluster,” *Frontiers in genetics*, vol. 9, p. 18, 2018.

[31] F. Almuqhim and F. Saeed, “Asd-saenet: a sparse autoencoder, and deep-neural network model for detecting autism spectrum disorder (asd) using fmri data,” *Frontiers in Computational Neuroscience*, vol. 15, p. 654315, 2021.

[32] X. Yang, M. S. Islam, and A. A. Khaled, “Functional connectivity magnetic resonance imaging classification of autism spectrum disorder using the multisite abide dataset,” in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2019, pp. 1–4.

[33] J. A. Nielsen, B. A. Zielinski, P. T. Fletcher, A. L. Alexander, N. Lange, E. D. Bigler, J. E. Lainhart, and J. S. Anderson, “Multisite functional connectivity mri classification of autism: Abide results,” *Frontiers in human neuroscience*, vol. 7, p. 599, 2013.

[34] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. Jakab, B. S. Khundrakpam, J. D. Lewis, Q. Li, M. Milham, *et al.*, “The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives,” *Frontiers in Neuroinformatics*, vol. 7, no. 27, p. 5, 2013.

[35] Y. Behzadi, K. Restom, J. Liau, and T. T. Liu, “A component based noise correction method (compcor) for bold and perfusion based fmri,” *Neuroimage*, vol. 37, no. 1, pp. 90–101, 2007.

[36] R. C. Craddock, G. A. James, P. E. Holtzheimer III, X. P. Hu, and H. S. Mayberg, "A whole brain fmri atlas generated via spatially constrained spectral clustering," *Human brain mapping*, vol. 33, no. 8, pp. 1914–1928, 2012.

[37] G. E. Hinton and R. Zemel, "Autoencoders, minimum description length and helmholtz free energy," *Advances in neural information processing systems*, vol. 6, 1993.

[38] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[40] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, 04 1993. [Online]. Available: https://doi.org/10.1093/clinchem/39.4.561

[41] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

BIOGRAPHICAL STATEMENT

Mihir Yashwant Ingole was born in Sakhardoh, Maharshtra, India, in 1998. He received his B.E. degree in Electronics and Telecommunication from Pune University, India, in 2019 and his Master of Science in Computer Science from The University of Texas at Arlington (UTA) in 2023. From 2019 to 2021, he worked as a Project Engineer in Wipro Ltd. In 2022, he joined Computer Science and Engineering department as a Linux Systems Administrator at UTA.

With his thesis, Mihir aims to make a contribution to the early diagnosis of developmental disorders, especially Autism Spectrum Disorder (ASD). By applying deep learning techniques to resting-state functional MRI data analysis, he aspires to improve the lives of individuals with ASD.