WISEN IT SOLUTIONS
Progressive Quality Thinking

Table of Contents

Abbreviation

ADHD	Attention deficit hyperactivity disorder
AENet	Auto-encoding Network

Title

Improving the Accuracy and Broad Applicability ADHS Detection using Contextual Learning

Chapter 1: Introduction

Abstract

Attention deficit hyperactivity disorder (ADHD) is a complex neurodevelopmental disorder that not only affects attention and behavior but is also intricately linked with sleep disturbances and immune system dysregulation. Recent research highlights that individuals with ADHD frequently experience sleep problems, which in turn exacerbate ADHD symptoms and contribute to cognitive and emotional difficulties. Immunological alterations, including elevated proinflammatory cytokines and hypothalamic–pituitary–adrenal axis dysfunction, have been observed among ADHD patients, suggesting a biological interplay between inflammation, sleep, and neurodevelopment. Deep learning (DL) methods have been increasingly applied to neuroimaging data to identify patients with psychiatric and neurological disorders.

In deep learning-based ADHD prediction and classification, enhancing the extraction of electroencephalogram (EEG) features is crucial for improving model accuracy. Traditional supervised learning methods rely on large, detailed annotated datasets, limiting the feasibility of large-scale training. Recently, self-supervised learning approaches using masking-and-reconstruction strategies have emerged, reducing dependence on labeled data. However, these methods are vulnerable to inherent noise and signal degradation in EEG data, which diminishes feature extraction robustness and overall model performance.

A graph neural network (GNN) can model and analyze the brain, imaging from morphology, anatomical structure, function features, and other aspects, thus becoming one of the best deep learning models in the diagnosis of ND. Some researchers have investigated the application of GNN in the medical field, but the scope is broad, and its application to NDs is less frequent and not detailed enough. This review focuses on the research progress of GNNs in the diagnosis of ND. Firstly, we systematically investigated the GNN framework of ND, including graph construction, graph convolution, graph pooling, and graph prediction. Secondly, we investigated common NDs using the GNN diagnostic model in terms of data modality, number of subjects, and diagnostic accuracy.

General System

The human brain undergoes remarkable changes from infancy through adolescence, shaping our cognitive, emotional, and behavioral traits. Neurodevelopment is a precisely timed and intricately orchestrated process, where perturbations and miscues in the formation of neural circuits can lead to neurodevelopmental disorders such as autism spectrum disorder (ASD) and attention-deficit/hyperactivity disorder (ADHD), which are often blanketed as “neurodivergence” in cognition and behavior. ADHD, which affects approximately 5% of childrenand 6.76% of adults, is diagnosed about twice as frequently in males as compared to females. These higher numbers are likely attributable to behavioral symptoms of ADHD being more easily recognizable in males than females. ADHD is generally characterized by inattentiveness, hyperactivity, and impulsivity. The Diagnostic and Statistical Manual of Mental Disorders 5 (DSM-5)defines the diagnosis of ADHD for children as the presence of at least 6 symptoms, including inattentiveness, hyperactivity, and impulsivity, whereas in adults this decreases to five such indicators. A fundamental question that remains unanswered is as to the underlying structural and functional anomalies in the brain that are likely responsible for various observed behavioral changes in ADHD. These alterations in the maturation of brain networks, notwithstanding synaptic plasticity, may underlie diverse behavioral traits, though the precise underlying causes of ADHD and other neurodevelopmental disorders, may they be genetic, cellular, molecular, or environmental, remain largely unknown. Moreover, various behavioral therapies and medications used to treat ADHD symptoms also remain controversial.This review will first compare ADHD symptoms in children and adults,

and how they relate to brain development during childhood and adolescence. Next, we will look at how neurotransmitters contribute to ADHD symptoms, and how stimulants and non-stimulants work to attenuate them. We will then discuss the mechanisms underlying the efficacy of ADHD medications and how they impact long-term cognitive function in both humans and preclinical models. Subsequently, we will review the implications of ADHD medications when taken during pregnancy, and the use of transcranial brain stimulation as a possible future treatment. Lastly, we will finish by proposing a potential paradigm shift which could involve integrating synaptic-level findings from animal models and linking them to more complex neuronal networks in humans to better understand the underlying causes of ADHD, and the impact of psychotherapies on long-term memory.

Attention-Deficit/Hyperactivity Disorder (ADHD) is one of the most common neurodevelopmental disorders, affecting millions of children and adults worldwide. It is characterized by persistent patterns of inattention, hyperactivity, and impulsivity that interfere with an individual's ability to function effectively in school, work, and social settings. While ADHD is often recognized in childhood, it is increasingly understood as a lifelong condition, with many individuals continuing to experience symptoms into adulthood. ADHD can present in various ways, depending on factors such as age, gender, and cultural background, making it a complex and multifaceted disorder to diagnose and treat.The significance of ADHD extends far beyond its prevalence. It impacts multiple areas of life, including academic achievement, job performance, relationships, and mental health. Children with ADHD are at higher risk for academic difficulties, disciplinary problems, and lower self-esteem, while adults with ADHD may struggle with employment, interpersonal relationships, and emotional regulation. Additionally, untreated ADHD can lead to comorbid conditions such as anxiety, depression, and substance use disorders, further complicating the lives of those affected. The societal impact of ADHD is also substantial, with considerable costs related to healthcare, education, and loss of productivity.Despite the growing awareness of ADHD, significant challenges remain in terms of diagnosis and treatment. ADHD is often underdiagnosed in certain populations, such as girls and minority groups, leading to disparities in access to care and treatment outcomes. Girls, for example, are more likely to present with inattentive symptoms, which are less noticeable than hyperactivity, often leading to delayed diagnosis or misdiagnosis. Similarly, ADHD in minority populations may be overlooked or attributed to other behavioral issues, resulting in fewer children and adults in these communities receiving the treatment they need

ADHD is characterized by a persistent pattern of inattention, hyperactivity, and impulsivity, which can significantly impact an individual's daily functioning and quality of life. The disorder typically manifests in early childhood and, if left untreated, can persist in adulthood. Despite its well-documented prevalence among children and adolescents, with an estimated global prevalence of 5.29%, ADHD is often underdiagnosed, particularly in adults. Many adults with ADHD were never evaluated or diagnosed as children, and they continue to struggle with the disruptive effects of the disorder throughout their lives. The symptoms of ADHD can manifest differently in adults compared to children. While hyperactivity may decrease with age, internal restlessness and difficulty with executive functions often persist. Adults with ADHD frequently struggle with time management, organization, and maintaining focus during important tasks. These challenges can affect various aspects of life, including career performance, relationships, and personal responsibilities. Many adults develop coping mechanisms to manage their symptoms, though these strategies may not always be effective.

ADHD has a strong heritable component, with genetic factors contributing approximately 74% to the overall risk of developing the disorder. Studies examining identical and non-identical twins have consistently shown that genetics plays a significant role in the inheritance of ADHD. However, large-scale genome-wide association studies (GWAS) have identified over 7300 genetic variants that increase risk, underscoring the polygenic nature of ADHD. Importantly, these variants alone do not guarantee the onset of ADHD, highlighting the need for an understanding of how environmental factors and gene–environment interactions contribute to the disorder's development.Key genes involved in dopamine regulation, such as DRD4 and DAT1, have emerged as central players in the genetic architecture of ADHD. These findings have been replicated across various studies, particularly through candidate gene and GWAS approaches, indicating cumulative small-risk variants that influence ADHD's development. Candidate gene studies have identified significant polymorphisms in genes like SLC6A3 (dopamine transporter), DRD4 and DRD5 (dopamine receptors), SLC6A4 (serotonin transporter), HTR1B (serotonin receptor), and SNAP25 (synaptosome protein). Both DRD4 and SNAP25 have shown strong links to ADHD, each being associated in over 70% of studies, while DRD5 and SLC6A3were associated in 60% of studies. Additionally, low activity of MAOA, a gene responsible for breaking down neurotransmitters like dopamine and serotonin, amplifies sensitivity to adverse environments, contributing to a higher risk for disorders such as ADHD and conduct disorders.Despite this, the overall effect size of individual variants is small, and much of the genetic risk is believed to arise from the cumulative influence of multiple small-risk variants, as indicated by polygenic risk scores (PRSs). This cumulative genetic risk suggests that genetic susceptibility interacts with environmental triggers to modulate ADHD onset, supporting the hypothesis that ADHD results from widespread dysfunction across brain networks rather than being confined to specific brain regions. The LPHN3 gene, for instance, has been linked to increased susceptibility to ADHD, while mitochondrial DNA haplogroups have also been implicated, suggesting a role for mitochondrial mechanisms in ADHD. The polygenic nature of ADHD reflects the contributions of multiple genes, each exerting small effects, but collectively influencing the risk for the disorder.Moreover, differential susceptibility theory proposes that certain genetic variations, known as "plasticity genes", increase an individual's sensitivity to environmental influences, whether positive or negative. This interaction between genetic predispositions and environmental exposures is critical to understanding the complex etiology of ADHD and its comorbidities. Understanding how both genetic and environmental factors converge can offer deeper insights into ADHD's development and

potential treatment strategies.

Attention deficit-hyperactivity disorder (ADHD) is a neurodevelopmental disorder with a high incidence in children associated with decreased levels of dopamine and norepinephrine in the brain. The current pharmacotherapy uses mainly inhibitors of dopamine and norepinephrine transporters such as the psychostimulants methylphenidate and amphetamine and the non-stimulant atomoxetine. ADHD is considered a neurodevelopmental disorder in a similar manner to autism spectrum disorder (ASD) and intellectual disability (ID). Early studies of postmortem brain samples of patients with ASD and ID showed an abnormal density of dendritic spines in cortical neurons associated with a putative defective or enhanced spine pruning during postnatal development.It is well known that the molecular machinery involved in synaptic transmission is enriched in the dendritic spines, that appear as small protuberances emerging from the dendritic shaft. These structures are remodeled during Hebbian and homeostatic plasticity events that modify the synaptic strength. The onset of symptoms associated with ASD, ID, and ADHD coincides with the peak of cortical spinogenesis occurring between late childhood and early adolescence. These data provide evidence for a neuroanatomical basis of these pathological states associated with alterations in dendritic spine maturation. Several experimental approaches such as in vivo high-resolution microscopy, electrophysiology and optogenetics have been used to study the activity-dependent remodeling of dendritic spines during synaptic plasticity processes in pyramidal neurons of the neocortex and hippocampus. Recently, several large-scale correlation analyses of neuroimaging and genetic data emerged from the Enhancing NeuroImaging Genetics through Meta-Analysis (ENIGMA) consortium and have allowed the identification of singularities and shared neuroanatomical alterations in neurodevelopmental and neuropsychiatric disorders such as ASD, ID, schizophrenia (SCZ) and ADHD

ADHD is a neurodevelopmental disorder with high prevalence (around 5%) among children worldwide characterized by hyperactivity, inattention and/or impulsivity affecting learning and sociability at school. Several studies support a polygenic cause for ADHD and non-genetic factors involved in the etiology of this disorder. Environmental factors such as smoking and alcohol consumption during pregnancy, exposure to contaminants such as lead, diet deficiencies and low educational attainment have been correlated with ADHD. In the 40–60% of children with ADHD, the disorder persists into adulthood. Furthermore, there is co-occurrence with psychiatric disorders such as depression and anxiety.Several neuroanatomical studies support the view that ADHD is a neurodevelopmental disorder. A significantly lower cortical thickness (most prominent in the prefrontal cortex, PFC) in children with ADHD as evaluated by brain MRI has been reported. The fitting of curves of thickness values obtained from sequential brain scanning in children diagnosed with ADHD and healthy controls show a delay of ≈3 years to attain the peak of cortical thickness during childhood, supporting a delay in cortical maturation in ADHD.Meta-analysis of neuroimaging data obtained in children, adolescents and adults with ADHD show a statistically significant reduction (compared to control groups) in cortical thickness, cortical surface area and volume of subcortical areas such as the accumbens, amygdala, caudate, putamen and hippocampus, in children with ADHD. Interestingly, the only brain area significantly reduced in adolescents is the hippocampus. Interestingly, analysis of brain MRI data from 145 groups of patients crossing six psychiatric disorders including ADHD, reveal a shared profile of differences in cortical thickness. It is noteworthy that the cross-disorder correlation analysis shows a positive value for differences in cortical thickness and gene expression between ADHD and major depressive disorder (MDD). Recently, a positive correlation between neuroanatomical parameters such as gray matter volume in neocortex and basal ganglia and the individual scores in N-back task (to evaluate working memory) in ADHD adolescents has been reported.Extensive evidence supporting a polygenic origin of ADHD with additional environmental risk factors has been reported. A recent genome-wide analysis has uncovered 12 significant risk loci in ADHD. The biological activities in these loci include neuronal differentiation (FOXP2), glutamate receptor trafficking and anchorage to postsynaptic density (SORCS3), regulation of dopamine transporter trafficking (DUSP6) and axon guidance (SEMA6D). The same research group presented an updated genome-wide association study (GWAS) increasing the number of ADHD cases included in the meta-analysis, identifying 27 risk loci including 6 of those previously reported by the same research group. Among these genes, mainly expressed in the frontal cortex (excitatory and inhibitory neurons), were found FOXP1 and FOXP2 (encoding transcription factors) previously involved in cannabis use disorder and intellectual disability . By another part, a familial ADHD phenotype caused by a missense mutation in the CDH2 gene, coding for the adhesion protein N-cadherin involved in synaptogenesis, has been recently described. Knock-in transgenic mice carrying this mutation exhibit behavioral and cognitive phenotypes associated with ADHD. In addition, CA1 pyramidal neurons contained in brain slices from these animals display a lower frequency of spontaneous miniature excitatory postsynaptic currents (mEPSC) without changes in the amplitude of currents, suggesting changes at the presynaptic level. Furthermore, CDH2-KO mice exhibit lower levels of tyrosine hydroxylase (TH) transcripts (qPCR) with a significant reduction in TH-positive neurons (immunohistochemical analysis). These results, considered together with the reduced dopamine concentration in PFC of mutant mice, support the effect of N-cadherin mutation on activity levels of dopaminergic pathways and its relevance for ADHD-associated neurophysiology.ADHD is associated with a dysfunction of dopamine and noradrenergic systems involving cortical areas such as the PFC (dorsolateral and ventromedial), cingulate cortex and basal ganglia (nucleus accumbens, caudate nucleus and putamen) affecting neural networks driving executive control, motor function, reward processing and decision-making. The pharmacological therapies approved by the American Food and Drugs Administration (FDA) for the treatment of ADHD include stimulants such as amphetamines and methylphenidate (MPH), and nonstimulants as atomoxetine (ATX). Amphetamines and MPH increase synaptic levels of dopamine and norepinephrine by inhibiting DAT and NET symporters, whereas ATX selectively inhibits the NET. The analysis of the dopamine levels in different brain areas by positron emission tomography (PET) imaging based on the competition between endogenous dopamine and

a synthetic radioligand of the D2/D3 dopamine receptor, show that MPH administration in ADHD patients decreases the radioactivity-associated signal in hippocampus to a lower magnitude compared to healthy controls. These data suggest a lower MPH-induced release of dopamine in the hippocampus of ADHD patients compared to healthy controls, suggesting an underactivation of dopamine pathways in this brain area. Considering drug efficacy and tolerability, the evidence that emerged from a network meta-analysis supports as first-choice medication, the use of methylphenidate in children and adolescents, and amphetamines in adults. Clonidine and guanfacine, activators of α2-adrenergic receptors, that are also used in ADHD pharmacotherapy, exhibit lower clinical efficacy in children (see Table 1). Among non-genetic factors involved in the development of ADHD are maternal smoking during pregnancy and exposure to environmental contaminants as lead and pesticides. Several animal models of ADHD using zebrafish, mice and rats have been developed allowing the study of cellular and molecular mechanisms underlying the action and efficacy of newly generated drugs for the treatment of the disorder. Rodent models of ADHD can be grouped into three types: genetic models such as SNAP-25 KO mice, DAT KO mice, LPHN-3 KO mice and rats, and those induced by exposure to environmental chemicals such as lead and pesticides, and by prenatal exposure to nicotine or alcohol. The prenatal nicotine exposure (PNE) murine model exhibits several symptoms described for ADHD patients, such as increased locomotor activity, low performance in cognitive tests, inattention, and transmissibility until the third generation. In addition, PNE mice exhibit a reduction in dopamine turnover in the frontal cortex and striatum and a significant volume reduction in the cingulate cortex

Attention deficit hyperactivity disorder (ADHD) is a neurodevelopmental pediatric-onset disorder with a prevalence of 5.29% worldwide. ADHD symptoms start to occur in childhood; however, some patients are diagnosed as adults, when the compensation mechanisms are no longer capable of allowing individuals to cope with the symptoms. Additionally, awareness about the disorder has significantly increased in recent decades, not only among doctors, but also among the general population, which has contributed to a greater number of ADHD diagnoses in the adult population. Although research on the topic indicates significant variability in the prevalence of ADHD, this phenomenon can be explained by methodological discrepancies and underdiagnosis in some populations. It is unlikely that the number of incidences has increased over the decades. According to the Diagnostic and Statistical Manual of Mental Disorders, 5th Edition, Text Revision, ADHD is a persistent pattern of inattention and/or hyperactivity-impulsivity that interferes with functioning or development; the main diagnostic criteria include inattention, hyperactivity, and impulsivity. The main types of the disorder are, as follows: predominantly inattentive; predominantly impulsive/hyperactive; and combined presentation. In general, possibly due to the influence of upbringing, there are gender differences in the clinical presentation—girls with ADHD were reported to exhibit more significant intellectual difficulties, exhibited less hyperactivity, and engaged in fewer externalizing behaviors. Patients with ADHD are more prone to experiencing difficulties with: quality of life (measured by lower life productivity, psychological health, and life outlook); relationships with peers (measured by lower level of the perceived quality of social relations in comparison to healthy peers through sociometric status)and family (depicted by lower Family Crisis Oriented Personal Evaluation Scale scores, thus greater family dysfunction); and school performance (including grade retention OR = 3.58, secondary school graduation failure OR = 2.41, or lower level diploma OR = 3.0). The disorder is marked by increased comorbidity with learning disorders and autism spectrum; autism spectrum and tic disorders; tic disorders and obsessive–compulsive disorder; obsessive–compulsive disorder and mood disorders; and mood disorders. The current treatment for ADHD disorder comprises both pharmacological and therapeutical approaches. Efficacious drugs used in the disorder include psychostimulants (methylphenidate, dexamphetamine, mixed amphetamine salts, and lisdexamfetamine) and non-psychostimulants (atomoxetine, viloxazine, bupropion). In contrast, non-pharmacological treatment consists of cognitive–behavioral therapy, neurofeedback training, and cognitive training interventions. Properly treated patients usually have alleviated symptoms of ADHD; however, treatment-related adverse side effects are common and inconvenient. The average ADHD treatment induces approximately six side effects [

Patients with ADHD constitute a heterogeneous group of people with different demands and needs with respect to improving their lives. One of the most prevalent problems (reported by 25–50% of patients with ADHD) includes sleep disturbances, and, thus, decreased quality of life, further impairment of cognitive performance, and more significant attention deficits. In recent years, researchers have investigated the connection between these symptoms and the immunological aspects of the ADHD profile; both share the same abnormalities in immunity and their influence on each other can be largely dependent on immunological processes.

In recent years, more research has explored the association between ADHD and aspects of immunology. In general, people with ADHD have altered immunological profiles, including typically higher levels of interleukin 6 (IL-6, known to influence microglia), IL-1-beta (a key proinflammatory cytokine known to influence microglia), IL-1-beta (a key proinflammatory cytokine), and tumor necrosis factor (TNF) alpha (related to a weak immune response), which negatively correlate with symptom severity. This means that ADHD subjects are more prone to developing immune disorders due to cytokine disruption. MicroRNA-200b-3p or taurine intake seems to lower those cytokines, which may be a promising pathway in the search for a treatment target. As ADHD is perceived as a neurodevelopmental disorder without a strictly defined genesis, the evaluation of particular microRNAs might be helpful in unraveling those biochemical relations. At a molecular level, ADHD contributes to the increase in myeloid dendritic cells, monocytes, and granulocytes. In the saliva of ADHD subjects, there were significantly elevated levels of alpha-amylase, immunoglobulin A, and immunoglobulin M, although no differences in morning cortisol were detected. However, according to a systematic review by Chang et al., ADHD individuals have lower cumulative cortisol with lower morning levels. This means that stress markers seem to be increased among ADHD subjects, pointing to a plausible role of such an enhanced immune response.

Children with ADHD were found to have a downregulated hypothalamic–pituitary–adrenal (HPA) axis, as shown in the presence of single nucleotide polymorphisms (especially rs9470080) in FK506 binding protein 5—involved in immunoregulation, e.g., via interactions with corticoid receptor complexes—and lower diurnal cortisol levels in comparison to healthy children. In addition, glucocorticoid receptors TthIIII_rs10052957, NR3C1-I_rs10482605, and ER22/23EK_rs6189/rs6190 variants were associated with ADHD diagnosis.

Both genetic and environmental factors contribute to shaping this profile. In terms of genetics, the offspring of ADHD parents are more prone to developing psychiatric disorders, including ADHD—up to 88% in terms of the formal heritability. In terms of environment, many risk factors have been suggested and confirmed to be associated with ADHD onset, including: maternal pre-pregnancy obesity, childhood eczema, hypertensive disorders during pregnancy and pre-eclampsia, maternal acetaminophen exposure during pregnancy, maternal smoking during pregnancy, childhood asthma, maternal pre-pregnancy overweight, and serum vitamin D. The influence of the mother's immunology was shown to be connected, especially in the context of the disorder's predisposition, as well as the offspring's immunology, as a triggering factor for ADHD occurrence. Environmental factors, such as increased air pollution, suggest that economically disadvantaged populations might be predisposed to the syndrome. Although many potential connections are not well-established, and the human immune system remains incompletely understood, some preliminary conclusions can be drawn

Differences can occur in the prenatal development of the immune system between subjects who will later develop neurodevelopmental disorders and those who will not. Although the heritability of ADHD is high, there are no confirmed genes, including proinflammatory genes, responsible for this. The most promising by far are those connected with dopamine and serotonin circuits. One polymorphism of dopamine transporter was proved to be associated with an almost three times greater prevalence of ADHD compared to children without this allele. The presence of dopamine receptor D4 polymorphism was shown to contribute to the increased risk of ADHD development. Both polymorphisms can be inherited but may also be influenced by prenatal smoking exposure. The serotonin-transporter-linked polymorphic region (5-HTTLPR)was found to be a bridge between maternal anxiety and negative emotionality, enhancing the vulnerability of the fetus to environmental risk factors for ADHD. Through the same polymorphism, maternal smoking might affect the fetal serotonin system, causing the fetus' vulnerability to emotional problems. Insights into the genetic risk of developing ADHD can be enhanced by an evaluation of pregnancy history. Both factors can cause innate disruption. Mothers with diverse inflammatory factors (such as obesity, autoimmune disease, infection during pregnancy, prolonged stress, smoking, and sleep disorders) are more likely to give birth to a child with subsequent ADHD. This process probably occurs, among other processes, through the activation of toll-like receptors, similar to other autoimmune disorders. In prenatal offspring, it induces neuroinflammation, affecting brain development and synaptic transmission, which is consistent with the neuroinflammation hypothesis in ADHD pathophysiology, suggesting that inflammation of nervous tissue can have a detrimental effect on the development of psychiatric disorders. As the nervous system starts its growth in the 3–4th week after conception, any unfavorable factor affecting neurogenesis and subsequent formations from this time have an impact on post-natal life, e.g., in forming significantly smaller total cerebral (lower by 3.2%) and cerebellar (lower by 3.5%) brain volume in ADHD children, with more detrimental changes among unmedicated ones (cerebral volume lower by 5.8%, cerebellar volume lower by 6.2%) based on magnetic resonance images. It is notable that similar changes occur in children and adults with ADHD; the extent of the morphologic changes correlate with specific subtypes of the disorder in ADHD children with more detrimental changes among unmedicated ones (cerebral volume lower by 5.8%, cerebellar volume lower by 6.2%) based on magnetic resonance images. In an animal model, the application of anti-inflammatory treatment to the mother ameliorated abnormal locomotor activity, indicating the protection of neurodevelopment and creating an opportunity to lessen the risk of ADHD occurrence in situations of excess maternal stress

Epilepsy is a common neurological disorder that affects approximately 50 million people worldwide. Patients with epilepsy experience recurrent seizures, which may lead to injuries, suffocation, or even death. Seizure detection helps accurately localize the epileptogenic zone (EZ), the brain region responsible for initiating seizures. Surgical resection of the EZ can render some patients seizure-free. Epilepsy can occur at any age, and early detection is crucial for preventing further damage during physiological development and improving patients' life expectancy. An electroencephalogram (EEG) is an objective method of recording brain activity using scalp electrodes. It can reveal abnormal brain activity and is widely used to study neuronal activity patterns associated with brain disorders. Deep learning algorithms have demonstrated potential in predicting seizures, offering the promise of improving quality of life for patients. However, current deep learning-based methods for epileptic seizure detection primarily rely on supervised learning strategies with datasets manually annotated by professionals, which poses several challenges in practical applications. First, acquiring large-scale, high-quality annotated data is not only time-consuming and labor-intensive, but also constrained by the scarcity of experts and the high annotation cost. Second, existing supervised training methods are often tailored to specific types of seizures, lacking generalization ability across diverse seizure patterns, which limits the generalizability and robustness of the algorithms. Additionally, existing methods fail to fully utilize the vast amount of unlabeled EEG data and face computational complexity challenges when processing large-scale, high-dimensional EEG signals. To address these issues, this paper proposes a self-supervised learning-based Transformer network for EEG signal analysis and epileptic seizure detection. This method reduces the reliance on annotated data and enhances the generalization ability for complex seizure patterns, providing technical support for early detection and precise prediction of epilepsy.

The human brain is a complex system containing approximately 100 billion neurons and trillions of synaptic connections. The brain's electrical activity became a research focus in the 19th century when Richard Caton recorded brain signals from rabbits. Brain recordings were also performed by Hans Berger, the first person to record electroencephalogram (EEG) signals from the human scalp. EEG-based research has since increased significantly, and EEG is now the most commonly used noninvasive tool to study dynamic signatures in the human brain. EEG signals measure voltage fluctuations at the scalp and reflect the instantaneous superposition of electric dipoles, primarily from dendritic inputs to large pyramidal cells in the neuropil. Signals traveling in white matter have traditionally been thought to be too fast to superimpose temporally, although recent cable theoretic modelsand empirical worksuggest that white matter may also contribute to brain rhythms measured at the scalp. Classically, the three primary forms of the brain's activity based on EEG signals are brain waves, event-related potential (ERP), and steady-state visual evoked potentials (SSVEPs). Among those, brain waves are most commonly used in EEG signal analysis for different types of tasks. Brain waves have been categorized in terms of five frequency bands: delta, 0.5–4 Hz; theta, 4–8 Hz; alpha, 8–13 Hz; beta, 13–30 Hz; and gamma, 30–150 Hz. Other classifications of brain signals can be found in previous publications.

Typically, EEG recordings were obtained by using an international 10–20 or 10–5 electrode placement system introduced by the American EEG Society. In this method, multiple noninvasive electrodes are placed on the surface of the scalp. Each electrode receives brain signals, which are then amplified and sent to a computer after being converted them into line images representing brain waves. Electrodes used in EEG recording can be categorized as wet electrodes, semi-dry electrodes, and dry electrodes. In wet electrode recording, conductive gel or paste, which is very uncomfortable for participants, must be placed between the electrodes and the skin. Semi-dry electrodes require a small amount of conductive gel, and dry electrodes do not require conductive gel or skin preparation. Despite its ease of use and quick setup, EEG recording from dry electrodes yields data with more artifacts than does gel-based EEG recording, thus affecting EEG analysis. In this case, identifying the most sensitive biomarkers for the specific task is essential. In addition, some experimental studies have confirmed that selecting the proper features can improve the performance of ML/DL methods. For instance, Dehais et al.have analyzed the performance of gel-free EEG recording with ERP and power spectral density (PSD) features along with the linear discriminant analysis (LDA) method and found that the performance of LDA with PSD features is much better than that of LDA with ERP. Thus, ERP appears to be more sensitive to noise than PSD features.

EEG is a low cost, noninvasive neuroimaging technique that provides high temporal resolution recordings of dynamic signatures in the brain; it has therefore become an indispensable tool in a variety of applications, including clinical diagnosis of a range of epileptic seizure types, brain-computer interface (BCI) systems, sleep analysis, and decoding behavioral activity. When interpreted carefully, classification tools can be used not only for prediction but also to gain neuroscientific knowledge. However, EEG signals are complex, high-dimensionaland non-stationary, and have a low signal-to-noise ratio in the temporal domain. Therefore, careful preprocessing is often required to remove artifacts, particularly when EEG data are collected concurrently and in the MRI setting.Although EEG has been demonstrated to be a valuable tool for research in various applications, it has several limitations, such as a low signal-to-noise ratio, nonlinearity and nonstationary properties, and inter-individual variabilitywhich affect analysis and processing performance. To address these limitations, EEG signal processing pipelines are often used. The general EEG classification pipeline includes data preprocessing, initializing the classification procedure, splitting the data set for the classifier, predicting the class of new data, and evaluating the classification model for the test data set. As shown in references, Riemannian geometry-based classifiers and adaptive classifiers have achieved success in classifying EEG signals. Machine learning (ML) and deep learning (DL) methods have become rapidly growing areas with applications in computational neuroscience, owing to higher levels of neural data analysis efficiency and decoding brain function. In ML and DL, various algorithms are used simplify processing pipelines and improve the learning process. For instance, supervised ML algorithms first learn on training data. The model and learned parameters are then applied to unseen or new data to predict the class label of the new data. Among different types of classification tasks, binary and multi-label classifications are widely used in clinical studies, and in studies of cognitive function, motor imagery (MI) processing, emotion recognition (ER), and brain disorders, including brain injury, attention disorders, and multiple sclerosis. In addition to these recognized areas, various computational neural models have been considered, such as the medial prefrontal cortex (mPFC) and anterior cingulate cortex in modeling of learning predictions and monitoring behavior. A recent computational model, the predicted response-outcome modelis based on the hypothesis that mPFC stores predictions of future outcomes.

Epilepsy is a neurological condition that affects a significant number of people worldwide and is characterized by recurrent seizures caused by abnormal electrical activity in the brain. Despite medical advancements, the diagnosis and identification of epilepsy remain challenging. However, electroencephalography (EEG) is a non-invasive technique that has been proven effective in detecting and diagnosing epilepsy by measuring the electrical activity of the brain through voltage differences between electrodes, providing both spatial and temporal information. With approximately 3 million adults and 470,000 children living with epilepsy in the United States alone, detecting this disorder is crucial in the field of medicine. Along with Alzheimer's disease, Parkinson's disease, multiple sclerosis, attention deficit hyperactivity disorder (ADHD), and migraine headaches, epilepsy is one of the most common neurological diseases that can affect people of all ages but is more prevalent in older adults. Neurological diseases are often chronic and progressive and significantly affect a person's quality of life and ability to function independently over time.Developing a reliable system for detecting epileptic seizures is a challenging task owing to the high variability of EEG signals, class imbalance problems, noise contamination, complexity of seizure patterns, and limited availability of data. The

variability in EEG signals depends on the individual's psychiatric and physiological conditions as well as the recording environment, which makes it difficult to create a generalized model that can perform well for various individuals and recording conditions. Seizures can also take different forms, including focal and non-focal, making the detection process more complicated. Additionally, signals arise from different types of neurons, resulting in issues of overlap and a lack of distinguishable features, which can further complicate the detection process. Human involvement in the detection process can introduce delays and errors, thereby reducing the effectiveness of the model. Furthermore, the number of seizure events in EEG recordings is relatively small compared to non-seizure events, leading to a class imbalance problem, where the model may not be able to accurately detect seizures owing to the scarcity of seizure samples in the training set. Moreover, EEG signals are often distorted by various types of noise, such as power line interference, muscle artifacts, and electrooculography (EOG) artifacts, which can undermine the accuracy of seizure detection in the presence of noise. Furthermore, seizures can manifest in diverse forms, making it challenging to design a system that can detect all types of seizures. Additionally, gathering a large quantity of EEG data for training and testing a seizure detection system is a time-consuming and difficult task. Nevertheless, significant progress has been made in the field of seizure detection in recent years, and future advancements are expected to lead to more reliable and precise seizure detection systems. Addressing these challenges in EEG signal analysis necessitates the application of advanced signal processing techniques, oversampling methods, and denoising procedures. Domain knowledge can also be integrated into this process. Deep learning algorithms, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTM), fully connected neural networks (FCNs), support vector machines (SVMs), and naive Bayes modelsare more resilient in handling the variability of EEG signals. They possess the capacity to learn intricate patterns and relationships within EEG signals, significantly contributing to their effectiveness in seizure detection

In recent years, deep learning (DL) methods have made significant strides in EEG analysis, with applications extending to epilepsy detection and classification. Convolutional neural networks (CNNs) have been widely employed for feature extraction due to their effectiveness in modeling spatial and temporal patterns of EEG signals. Notable examples include EEGNet, a lightweight architecture designed for brain–computer interface (BCI) applications, SCCNet, which integrates spatial and temporal convolutional modules to optimize spectral feature extraction, and FBCNet, which introduces spectral filtering during the initial feature extraction stage to enhance performance. Beyond CNNs, graph neural networks (GNNs) have shown promise in capturing the complex dependencies inherent in EEG data. Models like Time2Graph and SimTSC use graph structures to represent EEG signals, achieving outstanding results in tasks such as epilepsy diagnosis and emotion recognition. However, the reliance on predefined sensor topologies limits the generalizability of GNN-based approaches. More recently, Transformer-based models have attracted attention for their ability to capture long-term dependencies and global contextual features. For instance, EEGformer combines convolutional layers and Transformers to learn spatiotemporal features, MEET employs multi-band analysis to decode brain states, and ESTformer leverages spatiotemporal dependencies to enhance low-resolution EEG data. Despite significant advancements, most methods still rely on supervised learning, which requires costly and time-intensive large-scale annotated datasets. Furthermore, their reliance on task-specific annotations limits generalization and poses challenges in handling multi-task demands and complex clinical diagnoses.

Self-supervised learning (SSL) has emerged as a powerful tool in EEG signal analysis, addressing the challenges posed by limited labeled datasets. Self-supervised graph neural networks for improving EEG seizure analysis introduce a framework based on GNNs, where node prediction and edge reconstruction tasks are used to extract spatiotemporal dependencies among EEG channels, laying the foundation for utilizing graph structures in EEG analysis. Building on this, the BIOTmodel extends SSL to cross-dataset learning, leveraging channel segmentation and reassembly into "sentence"-like structures to handle heterogeneous biosignals and mismatched channel setups. This approach bridges the gap between fixed-structure GNNs and real-world EEG variability, emphasizing flexibility. Further enhancing universal EEG representation, EEGPTadopts a Transformer-based architecture with masked reconstruction and temporal alignment tasks, achieving robust performance across multi-task scenarios. This method integrates insights from BIOT's cross-domain focus but scales to broader, unified tasks using high-capacity Transformers.

Real Time Use Cases

ADHD has recently gained increasing attention in research focused on early diagnosis and prevention, and it is one of the most common neurodevelopmental disorders in childhood. It is characterised by symptoms of inattention, hyperactivity, and impulsivity that interfere with daily functioning and development. Globally, an estimated 6–10% of children and adolescents meet ADHD diagnostic criteria. Children with ADHD frequently encounter learning difficulties, impaired social communication, and elevated risks of mental health problems. These challenges often extend into adolescence and adulthood, affecting future educational achievement, psychosocial, and overall quality of life

However, traditional ADHD assessment methods can be broadly classified into clinical and non-clinical approaches. Clinical assessments, such as neuroimaging (e.g., magnetic resonance imaging), electroencephalography (EEG), and structured clinical interviews supplemented by standardised questionnaires, can provide relatively accurate diagnostic information and allow clinicians to evaluate symptom severity.

However, these methods are costly, time-consuming, and require specialised equipment and professional expertise, which limits their accessibility and large-scale use. Non-clinical approaches, by contrast, are mainly used for early screening and often rely on parent- or teacher-reported questionnaires. While these instruments are more feasible for population-level application, their accuracy may be compromised due to children's limited cognitive abilities and the potential biases or inconsistencies in parent and teacher reports

Motivation

ADHS, a common chronic brain disease, is caused by sudden excessive discharge of brain nerve cells. Nearly 1% of the world's population is suffering from epilepsy. During ADHDs, patients have transient involuntary convulsions in one part of the body (partial ADHDs) or the entire body (generalized ADHDs), sometimes accompanied by a loss of consciousness and urinary and fecal incontinence , greatly affecting the patients' life quality. The electroencephalogram (EEG) is a representative signal containing information on brain electrical activity, which is used as a tool for clinical diagnosis and analysis of epilepsy. Epilepsy is a chronic and recurrent brain dysfunction disease. An acute epileptic attack will interfere with a patient's normal behavior and consciousness, having a great impact on their life. The purpose of this study was to design a ADHD prediction model to improve the quality of patients' lives and assist doctors in making diagnostic decisions. This motivates us to develop this project.

Objectives

- To effectively capture both spatial and temporal patterns in EEG data
- To investigate how the class overlap problem influences the prediction performance on epilepsy ADHDs.
- To capture the representations that take into account the interactions among several variables at each point in time.
- To extract the features that would clearly distinguish the ADHD class while not taking into consideration extracting the discriminative features for the non-ADHD class as well.
- To perform the exponentialbased normalization process to eliminate the over-fitting problems in the output.
- To design using fewer internal layers than the other conventional deep learning architectures

Problem Statement

Two key components in research into ADHD detection and prediction using epileptic electroencephalography (EEG) signals are feature extraction and classification . Most of the existing research is patient-independent and trains models for all types of patients, while some EEG-based ADHD detection algorithms are patient-dependent and are adaptive to individual patients. In order to reduce the computational load for a real-time ADHD prediction using EEG data, identifying the most relevant channels for the ADHD prediction is both important and effective. It can make ADHD-predicting wearable or implantable devices with less complicated feature extraction during the process of developing machine learning algorithms for the real-time analysis. In addition, a decreased number of EEG channels may deliver more convenience to the patients.

Scope of the Project

The main contributions of this project are:

- Anaconda Installation and Configuration
- Data Analysis
- Data Preprocessing
- Training the Model
- Testing of Test Data

Domain Overview

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. Itâ€™s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

About Pandas

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandasâ€™ DataFrames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and DataFrames.

1. Series: a pandas Series is a one dimensional data structure (â€œa one dimensional ndarrayâ€) that can store values â€” and for every value it holds a unique index, too.
2. DataFrame: a pandas DataFrame is a two (or more) dimensional data structure â€“ basically a table with rows and columns. The columns have names and the rows have indexes.

Those who are familiar with R know the data frame as a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frameâ€™s rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc.

Now, DataFrames in Python are very similar: they come with the Pandas library, and they are defined as two-dimensional labeled data structures with columns of potentially different types.

In general, you could say that the Pandas DataFrame consists of three main components: the data, the index, and the columns.

Firstly, the DataFrame can contain data that is:

- a Pandas DataFrame
- a Pandas Series: a one-dimensional labeled array capable of holding any data type with axis labels or index. An example of a Series object is one column from a DataFrame.
- a NumPy ndarray, which can be a record or structured
- a two-dimensional ndarray

- dictionaries of one-dimensional ndarrayâ€™s, lists, dictionaries or Series.

Some of the key features of Python Pandas are as follows:

It provides DataFrame objects with default and customized indexing which is very fast and efficient.

- There are tools available for loading data of different file formats into in-memory data objects.
- It is easy to perform data alignment and integrated handling of missing data in Python Pandas.
- It is very simple to perform pivoting and reshaping of data sets in Pandas.
- It also provides indexing, label-based slicing, and sub-setting of large data sets.
- We can easily insert and delete columns from a data structure.
- ata aggregation and transformations can be done using group by.
- Data aggregation and transformations can be done using group by.
- High-performance merging and joining of data can be done using Pandas.
- It also provides time series functionality.
- Inserting and deleting columns in data structures.
- Merging and joining data sets.
- Reshaping and pivoting data sets.
- Aligning data and dealing with missing data.
- Manipulating data using integrated indexing for DataFrame objects.
- Performing split-apply-combine on data sets using the group by engine.
- Manipulating high-dimensional data in a data structure with a lower dimension using hierarchical axis indexing.
- Subsetting, fancy indexing, and label-based slicing data sets that are large in size.
- Generating data range, converting frequency, date shifting, lagging, and other time-series functionality.
- Reading from files with CSV, XLSX, TXT, among other formats.
- Arranging data in an order ascending or descending.
- Filtering data around a condition.
- Analyzing time series.
- Iterating over a data set.

Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just canâ€™t miss when youâ€™re learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices. To work with these arrays, thereâ€™s a vast amount of high-level mathematical functions operate on these matrices and arrays.

An array is basically nothing but pointers. Itâ€™s a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and
- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next row.

Or, in other words, an array contains information about the raw data, how to locate an element and how to interpret an element.

With NumPy, we work with multidimensional arrays. Weâ€™ll dive into all of the possible types of multidimensional arrays later on, but for now, weâ€™ll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with. In

fact, itâ€™s just a different way of thinking about a list of lists. A matrix has rows and columns. By specifying a row number and a column number, weâ€™re able to extract an element from a matrix.

We can create a NumPy array using the numpy.array function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. Because we want all of the elements in the array to be float elements for easy computation, weâ€™ll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats.

NumPy has several advantages over using core Python mathemtatical functions, a few of which are outlined here:

1. NumPy is extremely fast when compared to core Python thanks to its heavy use of C extensions.
2. Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library. Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.

NumPy comes with a variety of built-in functionalities, which in core Python would take a fair bit of custom code.

Mathplotlib

Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data vistualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams.

Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well.

Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB, if it is used in combination with Numpy and Scipy. Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB. This can make it extremely easy for MATLAB users to migrate to matplotlib.

Matplotlib can be used to create publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Another characteristic of matplotlib is its steep learning curve, which means that users usually make rapid progress after having started. The officicial website has to say the following about this: "matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code."

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks.

Deep learning algorithms are constructed with connected layers.

- The first layer is called the Input Layer
- The last layer is called the Output Layer
- All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.

A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.

To grasp the idea of deep learning, imagine a family, with an infant and parents. The toddler points objects with his little finger and always says the word 'cat.' As its parents are concerned about his education, they keep telling him 'Yes, that is a cat' or 'No, that is not a cat.' The infant persists in pointing objects but becomes more accurate with 'cats.' The little kid, deep down, does not know why he can say it is a cat or not. He has just learned how to hierarchies complex features coming up with a cat by looking at the pet overall and continue to focus on details such as the tails or the nose before to make up his mind.

A neural network works quite the same. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers.

The learning occurs in two phases.

1. The first phase consists of applying a nonlinear transformation of the input and create a statistical model as output.
2. The second phase aims at improving the model with a mathematical method known as derivative.

The neural network repeats these two phases hundreds to thousands of time until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration

Classification of Neural Networks

1. Shallow neural network: The Shallow neural network has only one hidden layer between the input and output.
2. Deep neural network: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers.

The computational models in Deep Learning are loosely inspired by the human brain. The multiple layers of training are called Artificial Neural Networks (ANN).

Neuron

Artificial Neural Networks contain layers of neurons. A neuron is a computational unit that calculates a piece of information based on weighted input parameters. Inputs accepted by the neuron are separately weighted.

Inputs are summed and passed through a non-linear function to produce output. Each layer of neurons detects some additional information, such as edges of things in a picture or tumors in a human body. Multiple layers of neurons can be used to detect additional information about input parameters.

Nodes

Artificial Neural Network is an interconnected group of nodes akin to the vast network of layers of neurons in a brain. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

Inputs

Inputs are passed into the first layer. Individual neurons receive the inputs, with each of them receiving a specific value. After this, an output is produced based on these values.

Outputs

The outputs from the first layer are then passed into the second layer to be processed. This continues until the final output is produced. The assumption is that the correct output is predefined.

Each time data is passed through the network, the end result is compared with the correct one, and tweaks are made to their values until the network creates the correct final output each time.

Some of the commonly used neural networks are as follows:

1. Artificial Neural Network (ANN)
2. Convolutional Neural Network (CNN)
3. Recurrent Neural Network (RNN)
4. Deep Neural Network (DNN)
5. Deep Belief Network (DBN)

Artificial neural networks are one of the main tools used in machine learning. As the â€œneuralâ€ part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

Generally, the working of a human brain by making the right connections is the idea behind ANNs. That was limited to use of silicon and wires as living neurons and dendrites.

Here, neurons, part of human brain. That was composed of 86 billion nerve cells. Also, connected to other thousands of cells by Axons. Although, there are various inputs from sensory organs. That was accepted by dendrites. As a result, it creates electric impulses. That is used to travel through the Artificial neural network. Thus, to handle the different issues, neuron send a message to another neuron.

As a result, we can say that ANNs are composed of multiple nodes. That imitate biological neurons of the human brain. Although, we connect these neurons by links. Also, they interact with each other. Although, nodes are used to take input data. Further, perform simple operations on the data. As a result, these operations are passed to other neurons. Also, output at each node is called its activation or node value.

Chapter 2: Literature Survey

Literature Survey 1	
Title	Quantitative Identification of ADHD Tendency in Children With Immersive Fingertip Force Control Tasks
Authors	Zhihao Zhang , Simin Kang , Jifan Yu , Heyang Li , Gaohan Yin , Hongxing Zhang , Li Sun and Dangxiao Wang
Published Year	2023
Efficiency	👍 Excellent performance in high dimension data processing, automatic feature extraction, and evolutionary learning ability. 👍 Lowering the Complexity Threshold 👍 Involves training multiple models in a sequence where each subsequent model aims to correct the errors of its predecessor.
Drawbacks	👎 The accuracy of the results of this method is limited by the precision and parameter selection of preprocessing and requires higher computational performance. 👎 Limited feature extraction methods, and lack of generalizability. 👎 This system is Opportunistic and uncontrollable
Description	Attention-deficit/hyperactivity disorder (ADHD) is a prevalent neurodevelopmental disorder that affects children. However, the traditional scale-based diagnosis methods rely more on subjective experiences, leading to a demand of objective biomarkers and quantified diagnostic methods. This study proposes a quantitative approach for identifying ADHD tendency based on fingertip pressing force control paradigm with immersive visual feedback. By extracting nine behavioral features from reaction time and dynamic force fluctuation features with high temporal and amplitude resolution, the proposed method can effectively capture the continuous changes in attention levels for ADHD diagnosis. The extracted features were analyzed using independent sample t-test and Pearson correlation to determine their association with ADHD-RS scale scores. Results showed that 12 statistical indicators were effective for distinguishing ADHD children from typically developed children, and several features of force control ability were also associated with core ADHD symptoms. A support vector machine (SVM) based classifier is trained for ADHD diagnosis and achieved an accuracy of 78.5%. This work provides an objective and quantitative approach for identifying ADHD tendency within a short testing time, and reveals the inherent correlation between the attention levels and the extracted features of reaction time and force fluctuation dynamics. This study explored the feasibility of using VR-based visual-haptic system to diagnose ADHD in children. We found significant differences in extracted behavior features between groups during the force control task, including differences in features and hyperactive and impulsive symptoms of ADHD- RS scale results, with fewer differences found regarding inattention symptoms. The SVM-based classification model achieved the accuracy of 78.5% in distinguishing ADHD children from typically developed children. In conclusion, the VR-based visual-haptic system has potential in the clinical application of ADHD tendency assessment. While this study represents a preliminary exploration of force control ability as a biomarker in identifying ADHD tendency, future studies must incorporate more neurobehavioral measures, such as eye tracking or electroencephalography, to continue developing the system in clinical settings. The authors thank Ziyuan Luo for contribution to this work, and they thank all the subjects for participating.

Literature Survey 2	
Title	Neuronal Correlates of Task Irrelevant Distractions Enhance the Detection of Attention Deficit/Hyperactivity Disorder
Authors	Chun-Chuan Chen , Eric Hsiao-Kuang Wu , Yan-Qing Chen , Ho-Jung Tsai , Chia-Ru Chung and Shih-Ching Yeh
Published Year	2023
Efficiency	👍 Outperforms single and conventional models, with a performance improvement. 👍 Best tuned parameters for Neural Network 👍 Provides the integrity and nontransferablity.
Drawbacks	👎 Cannot fully utilize global and local building prior information. 👎 Comes with huge computational cost as it consists of a huge number of parameters. 👎 Cannot meet current network business demands
Description	Early diagnosis and treatment can reduce the symptoms of Attention Deficit/Hyperactivity Disorder (ADHD) in children, but medical diagnosis is usually delayed. Hence, it is important to increase the efficiency of early diagnosis. Previous studies used behavioral and neuronal data during GO/NOGO task to help detect ADHD and the accuracy differed considerably from 53% to 92%, depending on the employed methods and the number of electroencephalogram (EEG) channels. It remains unclear whether data from a few EEG channels can still lead to a good accuracy of detecting ADHD. Here, we hypothesize that introducing distractions into a VR-based GO/NOGO task can augment the detection of ADHD using 6-channel EEG because children with ADHD are easily distracted. Forty-nine ADHD children and 32 typically developing children were recruited. We use a clinically applicable system with EEG to record data. Statistical analysis and machine learning methods were employed to analyze the data. The behavioral results revealed significant differences in task performance when there are distractions. The presence of distractions leads to EEG changes in both groups, indicating immaturity in inhibitory control. Importantly, the distractions additionally enhanced the between-group differences in NOGO $\alpha $ and $\gamma $ power, reflecting insufficient inhibition in different neural networks for distraction suppression in the ADHD group. Machine learning methods further confirmed that distractions enhance the detection of ADHD with an accuracy of 85.45%. In conclusion, this system can assist in fast screenings for ADHD and the findings of neuronal correlates of distractions can help design therapeutic strategies. In this study, we used an intelligent VR system integrated with EEG and machine learning methods to help diagnose chil- dren with ADHD. We first examined the differences in ERP and ERSP with and without distractions and found that the presence of distractions leads to 1) decreased P300 peak in the ADHD group, 2) reduced and ERSPs, and 3) prolonged N2 and P3 latencies. Furthermore, the distractions additionally enhanced the between-group differences in NOGO and ERSPs. By using machine learning methods, we validated that the presence of distractions can efficiently alter neuronal activities in children and improve detection of ADHD.

Literature Survey 3	
Title	Identifying ADHD for Children With Coexisting ASD From fNIRs Signals Using Deep Learning Approach
Authors	Jungpil Shin , Sota Konnai , Md. Maniruzzaman , Md. Al Mehedi Hasan , Koki Hirooka , Akiko Megumi and Akira Yasumura
Published Year	2023
Efficiency	Reduces the resources used for processing purpose. Has higher detection accuracy and is a lighter model for multi-scale detection in complex scenes. Effectively utilizes network parameters and reduces interference from background noise.
Drawbacks	Faces critical design challenges. Much depends on experts knowledge and experience. Generally have high polynomial running times.
Description	Attention deficit hyperactivity disorder (ADHD) for children is one of the most common neurodevelopmental disorders and its prevalence has increased globally. Children with ADHD are faced with various difficulties, including inattention, impulsivity, and hyperactivity. Therefore, it is important to use an early detection system that is simple, non-invasive, and automated. Children with ADHD also suffer from other coexisting one or more disorders, including major depressive disorder (MDD), autism spectrum disorder (ASD), etc and it creates more challenges to detect ADHD children. Very few researchers considered such kinds of these comorbidities in their studies to detect ADHD for children. In this work, we proposed a deep learning (DL)-based algorithm to identify ADHD children with coexisting ASD. Functional near-infrared spectroscopy (fNIRs) signals from thirteen ADHD children who have coexisting ASD and fifteen typically developing (TD) children were recorded during the drawing of handwriting patterns. We asked each child to draw periodic lines (PL) and zigzag lines (ZL) under the predict and trace condition and repeated them three times. Finally, a hybrid approach was designed by combining convolutional neural networks (CNN) and bidirectional long short-time memory (Bi-LSTM) to determine children with ADHD who have ASD. The experimental results showed that our proposed hybrid approach could determine ADHD children with coexisting ASD with a classification accuracy of 94.0%, a sensitivity of 89.7%, specificity of 97.8%, f1-score of 93.3%, and AUC of 0.938, respectively, for the PL predict task. In this paper, we proposed a DL-based approach for discrim- inating ADHD with coexisting ASD children from TD chil- dren using fNIRs signals. We asked each child (both ADHD J. Shin et al.: Identifying ADHD for Children With Coexisting ASD From fNIRs Signals Using DL Approach with coexisting ASD and TD) to draw two lines (PL and ZL) under the trace and predicted conditions. We employed three models (CNN, BiLSTM, and CNN-BiLSTM) to detect ADHD in children with ASD problems. Our experimen- tal results showed that the CNN-BiLSTM model achieved 94.0% accuracy for the PL line under predicted conditions. This result shows that the fNIRs signal is the most effec- tive biomarker for detecting ADHD comorbidity in chil- dren with ASD.

Literature Survey 4	
Title	Stability Analysis of fMRI BOLD Signals for Disease Diagnosis
Authors	Honorine Niyigena Ingabire , Haibo Qu , Min Li , Sixuan He , Joan Toluwani Amos , Yan Cui , Qing Wang , Dezhong Yao , Dan Ma and Peng Ren
Published Year	2022
Efficiency	It exploits global contextual information well with light and small computation. Ability To Deliver High Quality Results It provides easy information processing and cost reduction as well.
Drawbacks	Needs more formal statistical training. Requires more computing resources and increases the corresponding management cost. Approach is a bit time-consuming
Description	Previous studies have demonstrated that the stability changes in physiological signals can reflect individuals pathological conditions. Apart from this, according to system science theory, a large-scale system can generally be divided into many subsystems whose stability level govern its overall performance. Therefore, this study attempts to investigate the possibility of analyzing the stability of decomposed subsystems of resting-state fMRI (rs-fMRI) BOLD signals in order to assess the overall characteristic of the human brain and individuals health conditions. We used attention deficit/hyperactive disorder (ADHD) as an example to illustrate our method. Rs-fMRI BOLD signals were first decomposed into dynamic modes (DMs) which can illuminate the patterns of brain subsystems. Each DM is associated with one eigenvalue that determines its stability as well as oscillation frequency. Accordingly, we divided the DMs within common BOLD frequency bands into stable and unstable DMs. Then, the features related to the stability of those DMs were extracted, and nine common classifiers were used to differentiate healthy controls from ADHD patients taken from ADHD-200, a well-known dataset. The results showed that almost all features were statistically significant. Additionally, our proposed approach outperforms all existing methods with the highest possible precision, recall, and area under the receiver operating characteristic curve of 100%. In sum, we are the first to evaluate the stability of BOLD signals and demonstrate its possibility for disease diagnosis. This method can unveil new mechanisms of brain function, and could be widely used in medicine and engineering. According to previous studies, the stability of physiological signals generally reects an individuals health condition. The experimental results of this study conform to our hypothesis that the stability of fMRI BOLD signals can play an important role in disease diagnosis, which has not previously been demonstrated. We also showed the signicance of the system science perspective for analyzing fMRI BOLD signals, i.e. to regard BOLD signals as large-scale system and evaluating how the stability of their DMs (subsystems) affects its overall properties. Finally, the proposed approach has demonstrated the capability to provide signicant physiological information which are not given by existing methods. We have also demonstrated its great potential and accuracy in illuminat- ing individuals health conditions.

Literature Survey 5	
Title	Deep Spatio-Temporal Representation and Ensemble Classification for Attention Deficit/Hyperactivity Disorder
Authors	Shuaiqi Liu , Ling Zhao , Xu Wang , Qi Xin , Jie Zhao , David S. Guttery and Yu-Dong Zhang
Published Year	2021
Efficiency	👍 Robust and reliable neural network and the most standard machine learning algorithm that can be used on most types of prediction problems. 👍 Has gained popularity because of its end-to-end modeling, learning complex non-linear patterns, and automatic feature extraction abilities. 👍 Significantly improves the performance under severe occlusion conditions.
Drawbacks	👎 Not able to detect the convolution features on multiple scales by applying convolution kernel functions with different sizes and further realizes batch extraction of remote sensing image features. 👎 Feature maps with different resolutions are not aligned. 👎 May take huge time and economic cost to construct.
Description	Attention deficit/Hyperactivity disorder (ADHD) is a complex, universal and heterogeneous neurodevelopmental disease. The traditional diagnosis of ADHD relies on the long-term analysis of complex information such as clinical data (electroencephalogram, etc.), patients behavior and psychological tests by professional doctors. In recent years, functional magnetic resonance imaging (fMRI) has been developing rapidly and is widely employed in the study of brain cognition due to its non-invasive and non-radiation characteristics. We propose an algorithm based on convolutional denoising autoencoder (CDAE) and adaptive boosting decision trees (AdaDT) to improve the results of ADHD classification. Firstly, combining the advantages of convolutional neural networks (CNNs) and the denoising autoencoder (DAE), we developed a convolutional denoising autoencoder to extract the spatial features of fMRI data and obtain spatial features sorted by time. Then, AdaDT was exploited to classify the features extracted by CDAE. Finally, we validate the algorithm on the ADHD-200 test dataset. The experimental results show that our method offers improved classification compared with state-of-the-art methods in terms of the average accuracy of each individual site and all sites, meanwhile, our algorithm can maintain a certain balance between specificity and sensitivity. In this article, a new ADHD classication method based on fMRI is proposed, which can directly extract features from

Literature Survey 6	
Title	The Effects of Visual Stimuli on Attention in Children With Autism Spectrum Disorder: An Eye-Tracking Study
Authors	Bilikis Banire , Dena Al-Thani , Marwa Qaraqe , Kamran Khowaja and Bilal Mansoor
Published Year	2020
Efficiency	👍 Complex structures can be modelled more accurately. 👍 Can improve the worst-case performance 👍 It is a fast and easy procedure to perform
Drawbacks	👎 It is not an easy-to-use method 👎 High complexity, inaccuracy, and inadequacy 👎 Making it difficult to explain how the result was obtained.
Description	Attention is one of the fundamental elements of effective learning. The design of learning environments often consists of a blend of visual stimuli. Investigating the effect of visual stimuli types on the attention of children with autism spectrum disorder (ASD) is important for the theoretical understanding of attention. This study explores the effect of social and nonsocial visual stimuli on the attention of children with ASD and typically developing (TD) children in a simulated virtual classroom. Forty-six participants (ASD = 20, TD = 26) took part in a series of attention tests, in which social and nonsocial visual stimuli were used as target stimuli. We examined four eye-gaze measures: time to first fixate, first fixation duration, average fixation duration, and the sum of fixation count. The results show that social and nonsocial stimuli do not affect the attention of ASD and TD children. However, TD children pay significantly greater attention to target stimuli than children with ASD. These findings emphasize the strengths of children with ASD during attention tasks and the potential for the use of eye-gaze measures to identify attention impairment in children with ASD. This study thus recommends an investigation methodology for on-task attention assessment in a learning environment. This paper examined the effects of social and nonsocial stimuli on on-task attention in children with ASD and age- matched TD children. We analyzed four eye-tracking mea- sures (SFC, TTFF, AFD, and FFD) on target, social, and nonsocial visual stimuli. This study is one of the rst to investigate the effects of social and nonsocial visual stimuli on on-task attention in children with ASD using eye-gaze measures. Findings observed are as follows: (1) Children with ASD and TD children focus more on the target stimulus than social and nonsocial stimuli. This nding shows that social and nonsocial stimuli in the classroom setting do not affect their on-task attention. (2) Children with ASD show lower performance scores in the attention task, which emphasizes their impaired on-task attention. (3) The distraction type (static or dynamic) does not signif- icantly affect the performance scores of ASD and TD groups.

Literature Survey 7	
Title	The N270 in Facial S1-S2 Paradigm as a Biomarker for Children With Attention-Deficit/Hyperactivity Disorder
Authors	Fujun Zhao , Chao Yang and Yi Zheng
Published Year	2020
Efficiency	Better utilization of resources Works with very high degree of confidence. Reduces the resources used for processing purpose.
Drawbacks	Cannot can fit complex nonlinear relationships. There is no guarantee that a globally optimal boundary can be found. Have limitations in terms of processing speed and their inherent complexity prevents effective deployment.
Description	Children with attention-deficit/hyperactivity disorder (ADHD) might suffer from dysfunctions that affect their social interaction, however, this issue has been less investigated and the results were inconclusive, especially for face processing. Applying event-related potential technology, the current study used modified facial S1-S2 paradigm to investigate face recognition in children with ADHD and health control group. Twenty-nine ADHD children and twenty-nine health children were recruited. The results showed that central N270 was delayed under inconsistent condition than consistent condition for ADHD group and no significant difference was revealed between ADHD group and control group on amplitude and latency of N170, N270. N270 might be a sensitive neurophysiological marker for ADHD children.

Literature Survey 8	
Title	High-Accuracy Classification of Attention Deficit Hyperactivity Disorder With l2,1-Norm Linear Discriminant Analysis and Binary Hypothesis Testing
Authors	Yibin Tang , Xufei Li , Ying Chen , Yuan Zhong , Aimin Jiang and Chun Wang
Published Year	2020
Efficiency	👍 Has gained popularity because of its end-to-end modeling, learning complex non-linear patterns, and automatic feature extraction abilities. 👍 Simple to use and interpret 👍 Has higher detection accuracy and is a lighter model for multi-scale detection in complex scenes.
Drawbacks	👎 Cannot fully utilize global and local building prior information. 👎 This system is Opportunistic and uncontrollable 👎 Maximizes the complexity of the problem
Description	Attention Deficit Hyperactivity Disorder (ADHD) is a high incidence of neurobehavioral disease in school-age children. Its neurobiological diagnosis (or classification) is meaningful for clinicians to give proper treatment for ADHD patients. The existing ADHD classification methods suffer from two problems, i.e., insufficient data and noise disturbance. In this paper, a high-accuracy classification method is proposed by using brain Functional Connectivity (FC) as ADHD features, where an l2,1-norm Linear Discriminant Analysis (LDA) model and a binary hypothesis testing framework are effectively employed. In detail, we introduce a binary hypothesis testing framework to cope with insufficient data of ADHD database. The FCs of test data (without seeing its label) are used for training and thus affect the subspace learning of training data under binary hypotheses. On other hand, the l2,1-norm LDA model generates a subspace to represent ADHD features, aiming to overcome noise disturbance. By robustly learning ADHD features, the subspace energy difference between binary hypotheses becomes more discriminative. Thereby, the true hypothesis can be rightly estimated with its larger subspace energy, which provides reliable evidence to predict the label of test data. By the test on ADHD-200 database, it shows that our method outperforms other state-of-the-art methods with the significant average accuracy of 97.6%. Moreover, the corresponding result analysis with ADHD symptom score and the explanation of discriminative FCs between ADHD and healthy control groups are given, which further verifies the validity of our classification method. We propose a high-accuracy ADHD classication method. In this method, we use a binary hypothesis testing framework to effectively solve the insufcient data problem of ADHD databases. Meanwhile, an l2,1-norm LDA model is employed for the robust feature learning to alleviate noise disturbance. The experiments show our method signicantly outperforms the existing classication methods. The average accuracy achieves 97.6% with the LOOCV. Moreover, the correspond- ing analysis with ADHD symptom score and the explanation of discriminative FCs are also given, which well veries the validity of our classication method.

Literature Survey 9	
Title	Investigation of Attention Deficit/Hyperactivity Disorder Assessment Using Electro Interstitial Scan Based on Chronoamperometry Technique
Authors	Ree Nah Chua , Yuan Wen Hau , Chew Ming Tiew and Wan Leong Hau
Published Year	2019
Efficiency	Increased efficiency and speed Reducing the dissimilarity between the input data and the models depiction of the data. Works with very high degree of confidence.
Drawbacks	Narrowly specialized knowledge The gradient descent algorithm used can easily make the training result converge to the local minimum rather than the global minimum while ignoring the correlation between the local and the whole. Big payloads
Description	Attention Deficit Hyperactivity Disorder (ADHD) is a neurodevelopment disorder which causes inattention, hyperactivity, or impulsivity, that directly leads to the learning disability. The conventional ADHD assessment method such as behavioural rating scale requires long term behavior observation and subjective interpretation which may lead to misleading result. Over the past decades, there have been exponential growth in the research in biomedical engineering perspective such as Electroencephalograph (EEG), Magnetic Resonance Imaging (MRI) and electrodermal responding to identify accurate and objective measurements in ADHD assessment. Nevertheless, a clear definitive picture which link neuropsychological, behavioral, and neurophysiological research has yet to develop. The purpose of this study is to investigate ADHD in children aged from 7 to 14 years old at primary schools in Malaysia using Electro Interstitial Scan (EIS) technology which based on chronoamperometry technique. There are total of 182 children data have been collected, which consist of 58 ADHD children (study group) and 124 healthy children (control group). All subjects will undergo EIS measurement and the SPSS statistical analysis was conducted on the primary and secondary EIS parameters to identify significant level in differentiating ADHD and non-ADHD children. The statistical analysis result shows that EIS system reflects significant difference between control group and study group for both primary parameters (EIS channels 1, 2, 3, 4, 9, 10, 15, 16, 17 and 18) and secondary parameters (brain and neurotransmitter related parameters) (p <; 0.05). This study concludes that EIS system has good potential in providing a new measurable marker to complement the conventional assessment of ADHD as confirmatory decision support system. The aim of this study was to identify a novel assessment with objective measurement based on quantitative marker to support the conventional assessment of ADHD in children. This study concluded that the EIS system reect signicant difference between control group (normal healthy children) and study group (ADHD children) in primary as well as secondary brain and neurotransmitter related channels.

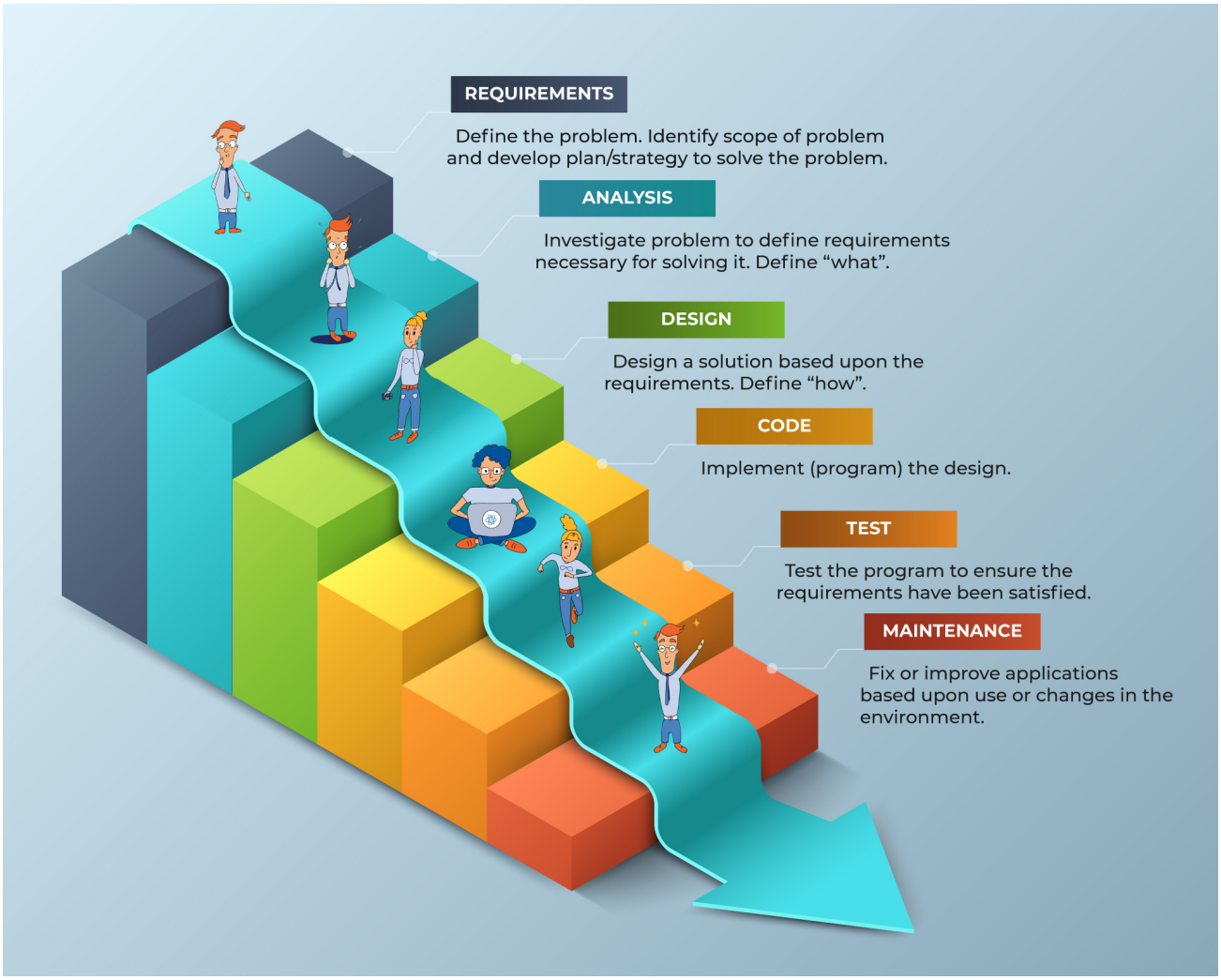
Literature Survey 10	
Title	3D CNN Based Automatic Diagnosis of Attention Deficit Hyperactivity Disorder Using Functional and Structural MRI
Authors	Liang Zou , Jiannan Zheng , Chunyan Miao , Martin J. Mckeown and Z. Jane Wang
Published Year	2017
Efficiency	Greatly improves the detection rate of hidden objects in complex environments. Improved reliability and resilience Boost the Performance
Drawbacks	Making it difficult to explain how the result was obtained. High complexity, inaccuracy, and inadequacy High computational complexity, sluggish processing speed, limited adaptability, and compromised robustness.
Description	Attention deficit hyperactivity disorder (ADHD) is one of the most common mental health disorders. As a neuro development disorder, neuroimaging technologies, such as magnetic resonance imaging (MRI), coupled with machine learning algorithms, are being increasingly explored as biomarkers in ADHD. Among various machine learning methods, deep learning has demonstrated excellent performance on many imaging tasks. With the availability of publically-available, large neuroimaging data sets for training purposes, deep learning-based automatic diagnosis of psychiatric disorders can become feasible. In this paper, we develop a deep learning-based ADHD classification method via 3-D convolutional neural networks (CNNs) applied to MRI scans. Since deep neural networks may utilize millions of parameters, even the large number of MRI samples in pooled data sets is still relatively limited if one is to learn discriminative features from the raw data. Instead, here we propose to first extract meaningful 3-D low-level features from functional MRI (fMRI) and structural MRI (sMRI) data. Furthermore, inspired by radiologists typical approach for examining brain images, we design a 3-D CNN model to investigate the local spatial patterns of MRI features. Finally, we discover that brain functional and structural information are complementary, and design a multi-modality CNN architecture to combine fMRI and sMRI features. Evaluations on the hold-out testing data of the ADHD-200 global competition shows that the proposed multi-modality 3-D CNN approach achieves the state-of-the-art accuracy of 69.15% and outperforms reported classifiers in the literature, even with fewer training samples. We suggest that multi-modality classification will be a promising direction to find potential neuroimaging biomarkers of neuro development disorders. With the availability of the large scale ADHD-200 dataset and the successes of deep learning in many recognition problems, we were motivated to develop an automatic classication algorithm based on deep learning to classify ADHD vs. TDC using MRI scans. Inspired by the way that radiologists exam- ine 3D brain images, we propose an automatic and effective 3D CNN architecture for ADHD classication which exploits the complementary information gleaned from both fMRI and sMRI. The proposed 3D CNN method is fundamentally dif- ferent from previous attempts to classify the ADHD using MRI scans.

Chapter 3: System Analysis

Waterfall Model

It is called as traditional approach. Waterfall is a linear method (sequential model) for any software application development. Here application development is segregated into a sequence of pre -defined phases.

Waterfall Model (Taken from Google.com)



1. Requirement gathering and documentation
   In this stage, you should gather comprehensive data approximately what this challenge requires. You may gather this data in a diffusion of ways, from interviews to questionnaires to interactive brainstorming. By means of the stop of this section, the task requirements have to be clean, and also you have to have a necessities file that has been allotted on your team.

2. System design
   Using the well-known requirements, your team designs the solutions. During this phase, no development will be happening. But the project team starts specification such as programming language or hardware requirements.

3. Implementation
   During this phase software development coding will be happening. Web application programmers take data from the previous stage and create a functional product. Web application programmers write source code in small pieces, which are integrated at the end of this phase or the beginning of the next.

4. Testing
   Once all coding is done, testing of the product can begin. Testers methodically find and report any problems. If serious issues arise, your project may need to return to phase one for revaluation.

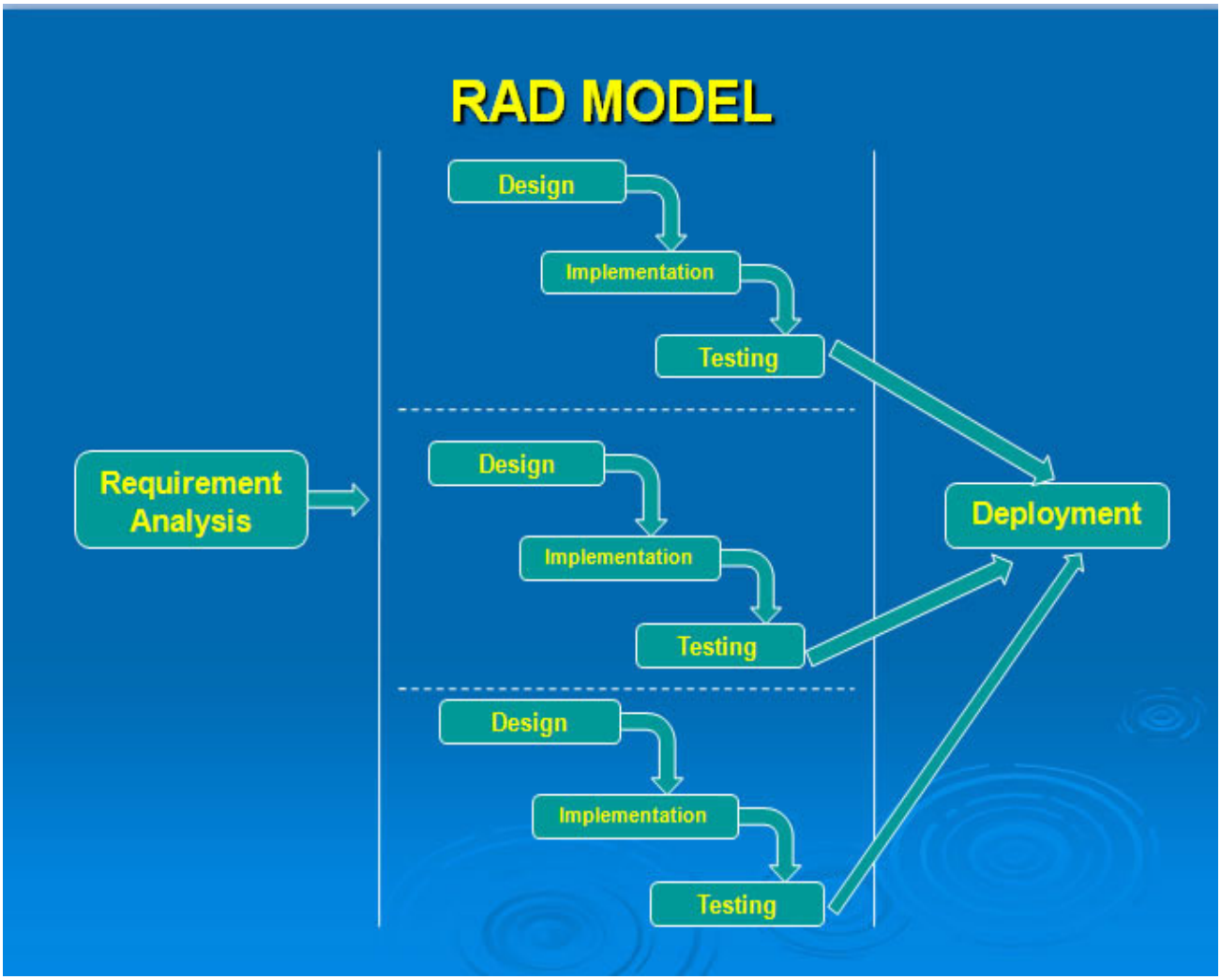
5. Delivery/deployment
   In this phase, the solution is complete, and your project team submits the deliverables to be deployed or released.

6. Maintenance
   The final solution has been implemented to the client and is being used. As troubles arises, the project team might also want to create patches and updates might also to deal with them. Again, huge troubles may also necessitate a return to segment one.

RAD Model

Rapid application development is an agile software development approach that focuses more on ongoing software projects and user feedback and less on following a strict plan. As such, it emphasizes rapid prototyping over costly planning.

RAD Model (Taken from Google.com)



1. Define Requirements
   Rather than making you spend months developing specifications with users, RAD begins by defining a loose set of requirements. â€œLooseâ€ because among the key principles of rapid application development is the permission to change requirements at any point in the cycle.

   Basically, developers gather the products gist. The client provides their vision for the product and comes to an agreement with developers on the requirements that satisfy that vision.

   This phase is equivalent to a project scoping meeting. Although the planning phase is condensed compared to other project management methodologies, this is a critical step for the ultimate success of the project.

   During this stage, web application programmers, clients (software users), and team members communicate to determine the goals and

expectations for the project as well as current and potential issues that would need to be addressed during the build.

1. Researching the current problem
2. Defining the requirements for the project
3. Finalizing the requirements with each stakeholders approval

A basic breakdown of this stage involves:

1. It is important that everyone has the opportunity to evaluate the goals and expectations for the project and weigh in. By getting approval from each key stakeholder and web application programmer, teams can avoid miscommunications and costly change orders down the road.

2. Prototype
   In this rapid application development phase, the developerâ€™s goal is to build something that they can demonstrate to the client. This can be a prototype that satisfies all or only a portion of requirements (as in early-stage prototyping).

   This prototype may cut corners to reach a working state, and thatâ€™s acceptable. Most RAD programming approaches have a finalization stage where developers pay down technical debt accrued by early prototypes.

   All the bugs and kinks are worked out in an iterative process. The web application programmer designs a prototype, the client (user) tests it, and then they come together to communicate on what worked and what did not.

   This method gives web application programmers the opportunity to tweak the model as they go until they reach a satisfactory design. Both the software web application programmers and the clients learn from the experience to make sure there is no potential for something to slip through the cracks.

3. Absorb Feedback
   With a recent prototype prepared, RAD developers present their work to the client or end-users. They collect feedback on everything from interface to functionalityâ€”it is here where product requirements might come under scrutiny.

   Clients may change their minds or discover that something that seemed right on paper makes no sense in practice. Clients are only human, after all. With feedback in hand, developers return to some form of step 2: they continue to prototype. If feedback is strictly positive, and the client is satisfied with the prototype, developers can move to step 4.

   Because the majority of the problems and changes were addressed during the thorough iterative design phase, web application programmers can construct the final working model more quickly than they could by following a traditional project management approach.

   The phase breaks down into several smaller steps:

   1. Preparation for rapid construction
   2. Program and application development
   3. Coding
   4. Unit, integration, and system testing

   The software development team of programmers, coders, testers, and web application programmers work together during this stage to make sure everything is working smoothly and that the end result satisfies the clients expectations and objectives. This third phase is important because the client still gets to give input throughout the process. They can suggest alterations, changes, or even new ideas that can solve problems as they arise.

4. Finalize Product
   During this stage, developers may optimize or even re-engineer their implementation to improve stability and maintainability. They may also spend this phase connecting the back-end to production data, writing thorough documentation, and doing any other maintenance tasks required before handing the product over with confidence.

Existing System

Attention deficit hyperactivity disorder (ADHD) is a prevalent neurodevelopmental disorder in children. Although numerous intelligent methods have been applied for its diagnosis, they seldom address symptom prediction, which is crucial for establishing the relationship between symptoms and subjective biosignals. We propose a severity prediction model, namely BSP-Net, which uses amplitude of low-

frequency fluctuation (ALFF) data and constructs severity predictors within a binary hypothesis testing (BHT) framework. Specifically, the existing system authors designed a dual-branch network for symptom severity prediction, with each branch corresponding to an assumed label for the test subject. Building on the accurate ADHD identification achieved by the existing auto-encoding network (AENet) within the BHT framework, the existing system authors integrate its network and introduce symptom score predictors in each branch. By comparing high-level features from both branches using the AENet, the existing system authors derive an estimated label. Once an assumed label is confirmed, the corresponding branchs score predictor is activated to generate the final symptom assessment. Using the BSP-Net, our experiments achieved severity prediction accuracies of 92.4% and 84.9% on specific ADHD-200 datasets, with a score tolerance threshold of 3. Moreover, the mean squared errors on these datasets were lower than 16, significantly outperforming other methods. Importantly, discriminative brain regions corresponding to typical ALFF data in the BSP-Net were identified as ADHD biomarkers. These biomarkers align with existing research on abnormal brain regions in children with ADHD. Consequently, our method demonstrates its validity by providing biological explanations derived from ADHD mechanisms. In this study, the existing system authors have developed an ADHD symptom severity assessment method, namely BSP-Net, leveraging a dual-branch network structure with ALFF data. Our approach addressed the limitations of capturing non-linear relationships between predictor variables and symptom scores, which are critical for accurate ADHD classification and biomarker detection. By integrating symptom score predictors within a binary hypothesis network, our method effectively estimated ADHD severity, where remarkable score prediction achieved 92.4% and 84.9% on the PU and NYU datasets, respectively, with the tolerance setting of 3. Moreover, our biomarker detection analysis identified several key brain regions highly associated with ADHD symptom severity. Through the alignment of these findings with existing literatures, the existing system authors further validated the reliability of our biomarkers, enhancing the explanatory power of our prediction model. Overall, the existsing BSP-Net not only improves the accuracy of symptom severity predictions but also strengthens the integrity of biomarker detection. Future work will focus on refining this method and exploring its applicability to severity assessment for ADHD subtypes and other neurological disorders, aiming to contribute to more precise and personalized diagnostic and therapeutic strategies.

Drawbacks of Existing System

- Leads to inaccurate ADHD prediction due to information loss and extends the warning time.
- The existing approach misidentifies some crucial outliers when dealing with objects.
- Does not explore the effectiveness of the existing algorithm on high-dimensional datasets.
- A model is vulnerable to overfitting, and significant changes will be required during training to avoid overfitting.
- Produce unsatisfactory outcomes due to insufficient ensemble design

Proposed System

We introduce an innovative circular graph visualization method for EEG data that provides a more intuitive and interpretable representation. This novel visualization facilitates effective classification into ictal and interictal classes, thereby achieving enhanced accuracy through visual inspection. This improvement significantly contributes to the practicality of ADHD detection. Our approach involves the aggregation of essential features, including frequency, statistical, and wavelet features, from all combined channels. This comprehensive method not only enriches the depth of the captured information but also substantially enhances the overall performance of our model.

The preprocessing step along with the module improve the ADHD detection performance. The hyperparameters used in the experiments were optimized for a particular patient. The same hyperparameters were then applied to the remaining patients to demonstrate the generalizability of our proposed approach. It should be noted that the performance can be enhanced further by optimizing the hyperparameters for each subject.Feature selection methods play a vital role in reducing computational complexity, improving computing times, and enhancing accuracy.During the training and evaluation of the classification models, data were min–max-normalized and a 5 × 2 cross-validation (2-fold, 5 repetitions) was implemented for model evaluation

Data decomposition, where the complete EEG data are sliced using a time window. A complete EEG signal is usually an EEG recording over a period of time, and slicing it makes it easier for subsequent processing. Classifying slices usually requires less hardware resources than classifying complete EEG data. In addition, a complete segment of EEG data usually contains both ADHD and nonADHD EEG signals, so it is less meaningful to classify the whole EEG signal. Classification of slices, on the other hand, allows for more accurate knowledge of whether a patient is having a ADHD at the moment corresponding to the slice.

Decomposing EEG data with a sliding time window, in addition to the length of the time window, affects the slicing results, and whether the

time windows overlap each other also affects the results. Compared with no overlap between time windows, the decomposition with an overlap can obtain more slices, which also leads to the local feature information being extracted multiple times.

When representing the slices generated from single-channel EEG signals, temporal information can be included in the images generated by the representation. When representing slices generated from multichannel EEG signals, the picture contains the information of each channel. In contrast, the one-dimensional representation contains less information, while the multidimensional representation contains more information, but often requires more computational resources;

To understand the role of GNNs in modeling ADHS detection, it is important to grasp the key components and operations that allow these models to process graph data effectively. We visualized the steps for analyzing brain data using graph neural networks.

Graph learning demonstrates powerful capabilities with broad implications for applications in domains such as social network analysis, recommendation systems, and bioinformatics. The EEG data collection process essentially maps brain activity onto a graph-like structure, where each node corresponds to an electrode location on the scalp, and edges represent the interactions and connectivity between different electrodes. By leveraging graph learning techniques adeptly, we can exploit the inherent characteristics of this graph structure to gain a more comprehensive understanding of brain activity patterns.

The graph-aggregation block is dedicated to enhancing and aggregating features from neighboring nodes. It employs GCNs with an attention mechanism to incorporate task-specific information into the graph. By doing so, discriminative features are amplified and effectively integrated. This step is crucial for capturing complex relationships and dependencies among nodes in the graph.In this final block, the fused features from different graph representations are combined and utilized for classification tasks. This achieves feature fusion across various graph representations and executes the final classification. This step ensures that the model effectively integrates information from different representations to make accurate predictions.

A general approach to EEG analysis using GNNs:

1. **Feature Extraction:** Extract features from the EEG signals recorded at each electrode. These features include amplitude, frequency-based features, and statistical measures.
2. **Data Representation:** EEG data can be represented as a graph, where nodes represent specific EEG channels, and edges represent the functional or structural connectivity between these channels.
3. **Graph Construction:** Construct a graph where electrodes are nodes, and the edges between nodes represent functional connectivity measures. The edges can be based on correlation, coherence, or other connectivity metrics derived from EEG signals.
4. **Graph Neural Network Architecture:** Design a GNN architecture that takes the EEG graph as input and processes it to extract relevant features and learn the underlying patterns in the connectivity data. GNNs can learn node and edge representations, capture spatial and temporal patterns, and model the dynamics of brain activity
5. **Training:** The Graph Neural Network: Train the GNN using labeled data, where the labels could represent different states or conditions.
6. **Validation and Testing:** Evaluate the trained GNN on unseen data to assess its performance in various EEG analysis tasks.
7. **Optimization:** Fine-tune the GNN architecture and hyperparameters to improve performance and efficiency

The main goals of the proposed model are as follows.

- To extract important features from the EEG signal that have the best properties for each signal, such as frequency content, patterns, and characteristics specific to ADHD, we extracted several frequency-based features, and statistics such as the mean, variance, skewness, and kurtosis were computed for each segment and employed as input features for the model. We employ a combination of wavelet and statistical features to analyze the EEG signal, as this multivariate approach was found to be more effective in detecting and diagnosing epilepsy than single-feature-based methods.
- Preprocessing of features was performed to generate minority class data, as ADHD windows typically contain a small number of samples.
-
- To evaluate the performance of our proposed model, we conduct a parametric comparison with previously published models for the detection of epilepsy from EEG signals using a set of standardized metrics, such as accuracy, sensitivity, and specificity. The results demonstrate the utility of utilizing both wavelet and statistical features in the detection and diagnosis of epilepsy from EEG signals.

Advantages of Proposed System

- Does not require the manual design of the corresponding feature extractor for each classification problem.
- Less prone to variations due to electrode location, physical characteristics.

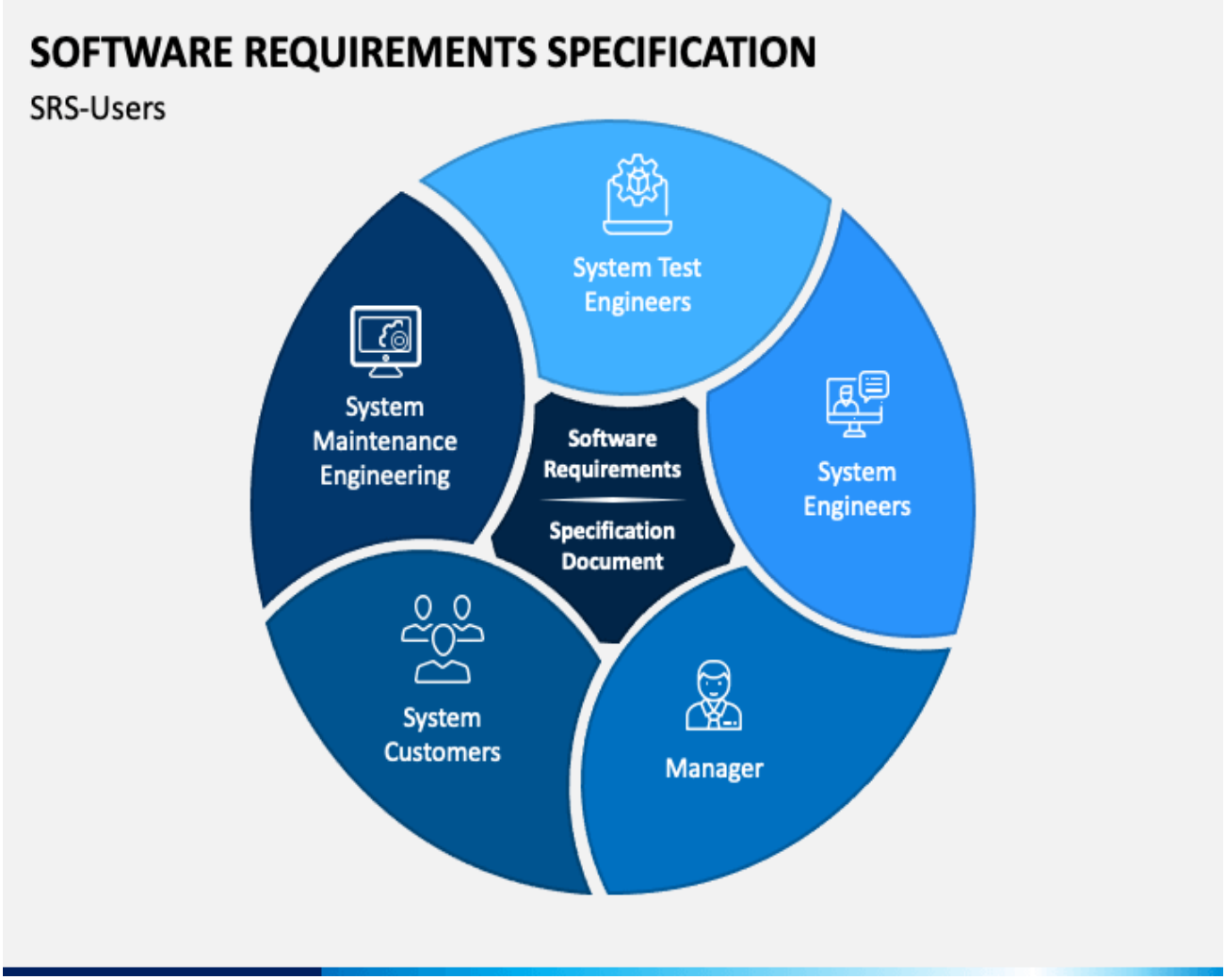
- Good generalization ability in the case of limited samples.
- Extract non-linear features and has good generalizing abilities.
- Does not require complex algorithms to process data

Chapter 4: Requirement Specification

Introduction

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need.

Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management.



A System Requirements Specification (SRS) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

Depending on the methodology employed (agile vs waterfall) the level of formality and detail in the SRS will vary, but in general an SRS should include a description of the functional requirements, system requirements, technical requirements, constraints, assumptions and acceptance criteria. Each of these is described in more detail below:

- Business Drivers
- Business Model
- Functional and System Requirements
- Business and System Use Cases
- Technical Requirements
- System Qualities
- Constraints and Assumptions

- Acceptance Criteria

Business Drivers

This section describes the reasons why the customer is looking to build the system. The rationale for the new system is important as it will guide the decisions made by the business analysts, system architects and developers. Another compelling reason for documenting the business rationale behind the system is that the customer may change personnel during the project. Documentation which clearly identifies the business reasons for the system will help sustain support for a project if the original sponsor moves on.

The drivers may include both problems (reasons why the current systems/processes are not sufficient) and opportunities (new business models that the system will make available). Usually a combination of problems and opportunities are needed to provide motivation for a new system.

Business Model

This section describes the underlying business model of the customer that the system will need to support. This includes such items as the organizational context, current-state and future-state diagrams, business context, key business functions and process flow diagrams. This section is usually created during the functional analysis phase.

Functional and System Requirements

This section usually consists of a hierarchical organization of requirements, with the business/functional requirements at the highest-level and the detailed system requirements listed as their child items.

Business and System Use Cases
This section usually consists of a UML use case diagram that illustrates the main external entities that will be interacting with the system together with the different use cases (objectives) that they will need to carry out. For each use-case there will be formal definition of the steps that need to be carried out to perform the business objective, together with any necessary pre-conditions and post-conditions.

The business use cases are usually derived from the functional requirements and the system use cases are usually derived from the system requirements.

Technical Requirements

This section is used to list any of the "non-functional" requirements that essentially embody the technical environment that the product needs to operate in, and include the technical constraints that it needs to operate under. These technical requirements are critical in determining how the higher-level functional requirements will get decomposed into the more specific system requirements.

System Qualities

This section is used to describe the "non-functional" requirements that define the "quality" of the system. These items are often known as the "-ilities" because most of them end in "ility". They included such items as: reliability, availability, serviceability, security, scalability, maintainability.

Constraints and Assumptions

This section will outline any design constraints that have been imposed on the design of the system by the customer, thereby removing certain options from being considered by the developers. Also, this section will contain any assumptions that have been made by the requirements engineering team when gathering and analyzing the requirements. If any of the assumptions are found to be false, the system requirements specification would need to be re-evaluated to make sure that the documented requirements are still valid.

Acceptance Criteria

This section will describe the criteria by which the customer will "sign-off" on the final system. Depending on the methodology, this may happen at the end of the testing and quality assurance phase, or in an agile methodology, at the end of each iteration.

The criteria will usually refer to the need to complete all user acceptance tests and the rectification of all defects/bugs that meet a pre-determined priority or severity threshold.

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems.

Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.

Python coding style comprises physical lines as well as logical lines or statements. A physical line in a Python program is a sequence of characters, and the end of the line terminates the line sequence as opposed to some other languages, such as C and C++ where a semicolon is used to mark the end of the statement. A logical line, on the other hand, is composed of one or more physical lines. The use of a semi-colon is not prohibited in Python, although itÃ¢â‚¬â„¢s not mandatory. The NEWLINE token denotes the end of the logical line. A logical line that only contains spaces, comments, or tabs are called blank lines and they are ignored by the interpreter.

As we saw that in Python, a new line simply means that a new statement has started. Although, Python does provide a way to split a statement into a multiline statement or to join multiple statements into one logical line. This can be helpful to increase the readability of the statement. Following are the two ways to split a line into two or more lines:

- **Explicit Line Joining**
  In explicit line joining, we use a backward slash to split a statement into a multiline statement.

- **Implicit Line Joining**
  Statements that reside inside [], {}, or () parentheses can be broken down into two or more physical lines without using a back slash.

Multiple Statements on a Single Line

In Python, it is possible to club multiple statements in the same line using a semi-colon; however, most programmers do not consider this to be a good practice as it reduces the readability of the code.

Whitespaces and Indentation

Unlike most of the programming languages, Python uses indentation to mark a block of code. According to Python coding style guideline or PEP8, we should keep an indent size of four.

Most of the programming languages provide indentation for better code formatting and do not enforce to have it. But in Python it is mandatory. This is why indentation is so crucial in Python.

Comments in any programming language are used to increase the readability of the code. Similarly, in Python, when the program starts getting complicated, one of the best ways to maintain the readability of the code is to use Python comments. It is considered a good practice to include documentations and notes in the python syntax since it makes the code way more readable and understandable to other programmers as well, which comes in handy when multiple programmers are simultaneously working on the same project.

Following are different kinds of comments that can be included in our Python program:

- Single Line Comments
  Single line Python comments are marked with # character. These comments end at the end of the physical line, which means that all characters starting after the # character (and lasts till the end of the line) are part of the comment.

- Docstring Comments
  Python has the documentation strings (or docstrings) feature which is usually the first statement included in functions and modules.

  Rather than being ignored by the Python Interpreter like regular comments, docstrings can actually be accessed at the run time using the dot operator.

  It gives programmers an easy way of adding quick notes with every Python module, function, class, and method. To use this feature, we use triple quotes in the beginning of the documentation string or comment and the closing triple quotes at the end of the documentation comment. Docstrings can be one-liners as well as multi-liners.

- Multiline Comments
  Unlike some programming languages that support multiline comments, such as C, Java, and more, there is no specific feature for multiline comments in Python. But that does not mean that it is totally impossible to make multiline comments in Python. There are two ways we can include comments that can span across multiple lines in our Python code.

Python Block Comments: We can use several single line comments for a whole block. This type of comment is usually created to explain the block of code that follows the Block comment. Python Block comment is the only way of writing a real comment that can span across multiple lines. It is supported and preferred by Pythons PEP8 style guide since Block comments are ignored by Python interpreter or parser.

Data Types

One of the most crucial part of learning any programming language is to understand how data is stored and manipulated in that language. Users are often inclined toward Python because of its ease of use and the number of versatile features it provides. One of those features is dynamic typing.

In Python, unlike statically typed languages like C or Java, there is no need to specifically declare the data type of the variable. In dynamically typed languages such as Python, the interpreter itself predicts the data type of the Python Variable based on the type of value assigned to that variable.

Advantages of Python

- Universal Language Construct
- Support both High Level and Low Level Programming
- Language Interoperability
- Fastest Development life cycle therefore more productive coding environmentLess memory used because a single container hold
- Multiple data types and each type doesnÃ¢â‚¬â„¢t require its own function
- Learning Ease and open source development
- Speed and user-friendly data structure
- Extensive and extensible libraries.
- Simple & support IoT
- and many more

Anaconda Software

Anaconda is the data science platform for data scientists, IT professionals and business leaders of tomorrow. It is a distribution of Python, R, etc. With more than 300 packages for data science, it becomes one of the best platforms for any project.

Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to easily collect data from various sources using various machine learning and AI algorithms. It helps in getting an easily manageable environment setup which can deploy any project with the click of a single button.

Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms.

Anaconda Navigator is a desktop GUI that ships with Anaconda and lets you launch applications and manage conda packages, environments, and channels without having to use a command-line interface. It can search for packages in a local Anaconda repository or on Anaconda Cloud. With Navigator, you donâ€™t need to type commands in a terminal, it lets you work with packages and environments with just a click.

Jupyter Notebook

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. In other words: itâ€™s a single document where you can run code, display the output, and also add explanations,

formulas, charts, and make your work more transparent, understandable, repeatable, and shareable.

Using Notebooks is now a major part of the data science workflow at companies across the globe. If your goal is to work with data, using a Notebook will speed up your workflow and make it easier to communicate and share your results.

As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet.

Its two main components are the kernels and a dashboard.

- A kernel is a program that runs and introspects the users code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages.
- The dashboard of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary.

Jupyter Notebook Features

- **Pluggable authentication**
  Manage users and authentication with PAM, OAuth or integrate with your own directory service system.
- **Centralized deployment**
  Deploy the Jupyter Notebook to thousands of users in your organization on centralized infrastructure on- or off-site.
- **Container friendly**
  Use Docker and Kubernetes to scale your deployment, isolate user processes, and simplify software installation.
- **Live coding environments**
  Code can be changed and run in real-time with feedback provided directly in the browser
- **Code meets data**
  Deploy the Notebook next to your data to provide unified software management and data access within your organization.

TensorFlow

Deep learning is a subfield of machine learning that is a set of algorithms that is inspired by the structure and function of the brain. Deep learning is a subset of machine learning. There are certain specialties in which we perform machine learning, and that's why it is called deep learning. For example, deep learning uses neural networks, which are like a simulation of the human brain. Deep learning also involves analyzing large amounts of unstructured data, unlike traditional machine learning, which typically uses structured data. This unstructured data could be fed in the form of images, video, audio, text, etc.

TensorFlow is the second machine learning framework that Google created and used to design, build, and train deep learning models. You can use the TensorFlow library do to numerical computations, which in itself does not seem all too special, but these computations are done with data flow graphs. In these graphs, nodes represent mathematical operations, while the edges represent the data, which usually are multidimensional data arrays or tensors, that are communicated between these edges.

The name TensorFlow is derived from the operations which neural networks perform on multidimensional data arrays or tensors! Its literally a flow of tensors.

Using tf.keras allows you to design, fit, evaluate, and use deep learning models to make predictions in just a few lines of code. It makes common deep learning tasks, such as classification and regression predictive modeling

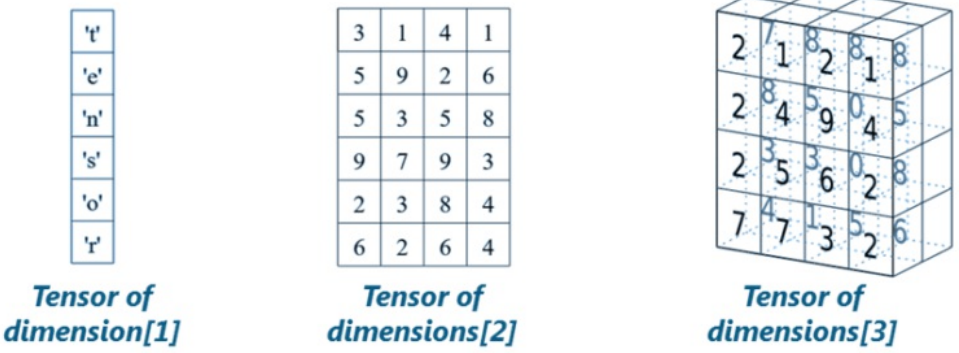
The other important aspect is TensorFlow is highly scalable. You can write your code and then make it run either on CPU, GPU, or across a cluster of these systems for the training purpose.

Generally, training the model is where a large part of the computation goes. Also, the process of training is repeated multiple times to solve any issues that may arise. This process leads to the consumption of more power, and therefore, you need a distributed computing. If you need to process large amounts of data, TensorFlow makes it easy by running the code in a distributed manner.

GPUs, or graphical processing units, have become very popular. Nvidia is one of the leaders in this space. It is good at performing mathematical computations, such as matrix multiplication, and plays a significant role in deep learning. TensorFlow also has integration

with C++ and Python API, making development much faster.

A tensor is a mathematical object represented as arrays of higher dimensions. These arrays of data with different sizes and ranks get fed as input to the neural network. These are the tensors.



You can have arrays or vectors, which are one-dimensional, or matrices, which are two-dimensional. But tensors can be more than three, four or five-dimensional. Therefore, it helps in keeping the data very tight in one place and then performing all the analysis around that.

Hardware & Software Requirements

Hardware Requirements

- Processor: Minimum i5 Dual Core
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 200 GB; Recommended 200 GB or more
- Memory (RAM): Minimum 16 GB; Recommended 32 GB or above

Software Requirements

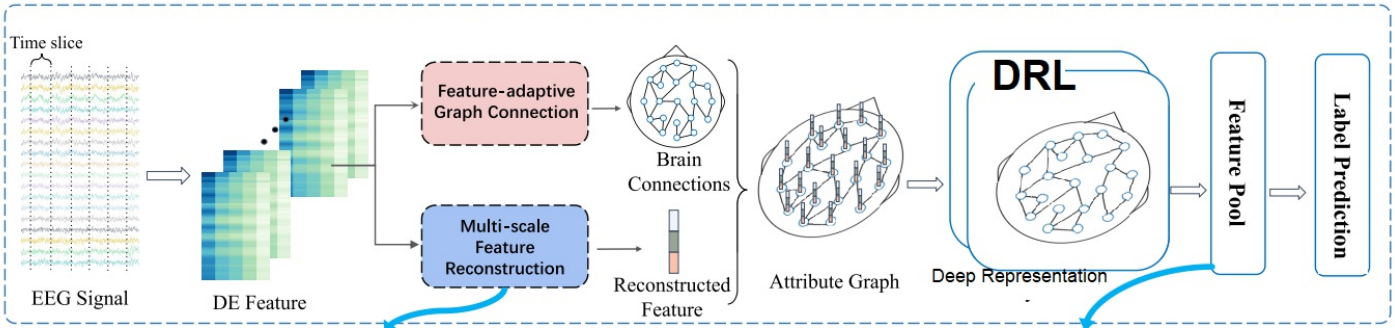
- Python For AI/ML/DL Programming
- Jupyter Notebook IDE (Integrated Development Environment) for Development.
- PyTorch or TensorFlow for Deep Learning Coding.
- Sklearn for Machine Learning/Feature Extraction/Evaluation Metrics Coding.
- Numpy for implementing Linear Algebra.
- Plotly for Data Visualization (For Graphs).
- Matplotlib for Data Visualization (For Graphs).
- Seaborn for Data Visualization (For Graphs).
- Pandas for dealing with Tabular Data.

Chapter 5: System Design

A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Mainly, UML has been used as a general-purpose modelling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency.

Architecture



Existing Algorithm

- Auto Encoding Network

Proposed Algorithm

- Graph Neural Network Learning Model

Advantages of Proposed Algorithm

Graph Neural Network Learning Algorithm Advantages Advantages

- High accuracy. Modeling complex relationships.
- Easier to use with less need for tweaking parameters.
- Combines several statistical approaches to improve the final predictive performance.

Chapter 5: System Implementation

Project Modules

Module 1 : Preprocessing

Epileptic EEG signals can be divided into three stages, namely, interictal, preictal, and ictal. In our ADHD prediction task, we mainly focused on the preictal and interictal periods. Therefore, we discarded the ictal EEG segments and transformed ADHD prediction into a binary classification problem. At the same time, we rearranged the EEG data for each patient according to the electrode order. In this dataset, the interictal and preictal data were severely unbalanced since some patients had fewer ADHDs during the monitoring period. In order to overcome this problem, we used overlapping sliding windows in the training stage to obtain more preictal fragments. To ensure that the ratio of the two types of training data was close to 1, we set the window size of preictal to W_p as follows:

$$W_p = \frac{n_p}{n_i} W$$

where n_p and n_i represent the number of EEG segments in each patient's preictal and interictal stage, respectively, and represents the size of the EEG window.

Module 2 : Feature Extraction

To extract rich features, a bandpass filter was employed to transfer each EEG segment into five physiological frequency bands.

There are three types of brain connections : (1) structural connectivity, an anatomical connection between brain neurons; (2) functional connectivity (FC), a statistical interdependence between different neuronal activities, which belongs to undirected connection; (3) effective connectivity (EC), the causal effect of one neural region on another, which belongs to directed connection. Only FC and EC were calculated in each band because structural connectivity cannot be obtained using EEG data. We selected measures based on different mathematical assumptions to obtain various adjacency matrices of a brain network.

Module 3 : Classification Methods - ML

Decision tree (DT): This is a hierarchical model composed of decision nodes and terminal leaves, each leaf have an output label. Decision nodes implement a test function () , which is a discriminator dividing the input space into smaller regions. Among all possible splits, the DT looks for the one that minimizes impurity. There are several impurity measures, e.g., the Gini index and entropy. For a two-class problem, the Gini index is defined as

$$\phi\left(p,\ 1-p\right) = 2p\left(1-p\right)$$

where p is the probability of a sample reaching a node m, to belong to a class C. The classification and regression trees algorithm (CART) was applied in this research.

Support vector machine (SVM): This constructs a hyperplane or set of hyperplanes in a high-dimensional space that can be used for classification. Those hyperplanes have the largest distance to the nearest training data points (also known as the functional margin) . The task of finding the optimal separating hyperplanes can be defined as

$$min\frac{1}{2}||w||^2 \; subject\ to\ r^t \left(w^T x^t + w_0 \right) \geq 1, \; \square t$$

where w are the parameters defining the hyperplane, x_t are the instances of the training set, and r_t is the actual label. If the problem is not linearly separable, the problem can be mapped to a new space by using non-linear basis functions

K-nearest neighbor (KNN): This is a classifier that learns by analogy. A target unknown instance is compared to all the instances in the training set, locating the k closest instances; the algorithm assigns the class that corresponds to the majority. “Closeness” is measured by using a distance metric; in this study, we used the Manhattan distance (selected through a parameter grid search). The Manhattan distance for a p-dimension space is defined as

$$d\left(i,j\right) = \left|x_{i1} - x_{j1}\right| + \left|x_{i2} - x_{j2}\right| + \ldots + \left|x_{ip} - x_{jp}\right|$$

Random forest (RF): This is an ensemble method, and each classifier of the ensemble is a DT. For each node of the DT, a random selection of features is used to determine the best split. RF also uses bagging (bootstrap aggregation), which means that the training set for each DT is sampled with replacement from the original training set. After model training, each DT votes, and the most voted class is assigned to the test instance

Module 4 : Classification Methods - GNN

A graph neural network (GNN) is an extension of an artificial neural network that is capable of learning on graph-structured data. Specifically, we implement a graph convolutional network (GCN) for a graph classification task. Graph neural networks (GNNs) are specialized neural networks designed to operate on graph data structures, predictingnodes, edges, and graph-based tasks. They are particularlyeffective for tasks like graph classification and utilizing EEGrecordings to construct graphs and make predictions. GNNsextend traditional neural network operators to graphs, lackinga clear sense of distance or order like images, with graphconvolutional layers being a key component. These layersemploy a message-passing mechanism, updating each node’srepresentation by aggregating messages from neighboringnodes. This methodology is analogous to image convolution, allowing the features of nodes to disseminate acrossthe graph effectively

The input to the GCN classifier is in the form of a graph: G={N,E,F} , where N , E , and F are sets of nodes, edges and node features, respectively. The nodes are fixed in our case, as this is the number of EEG electrodes. The set of edges E is given by the adjacency matrix A computed by the FC measures introduced in the previous section. Finally, the node feature matrix F is an N×D matrix where each row encodes a D -dimensional feature for the corresponding node. Specifically, power spectral density (PSD) is computed over 1 Hz increments in an interval between 0 and 100 Hz, forming a 100-dimensional node feature vector

GCN is based on the message-passing framework, which assumes that neighbouring nodes should have similar node features. Briefly, a GCN layer updates the node features (i.e. messages) using the optionally transformed messages collected from neighbouring nodes. On a node level, a single GCN layer effectively aggregates information from the 1-hop neighbourhood of each node. Thus, stacking L GCN layers represents aggregation from L -hop neighbourhood. Formally, the GCN layer is implemented on a node-level as follows

$$\boldsymbol{x}_{\mathrm{i}}^{\mathrm{l}} = \boldsymbol{\Theta}_1 \boldsymbol{x}_{\mathrm{i}}^{\mathrm{l}-1} + \Theta_2 \overset{\max}{\mathrm{j}\in \mathrm{G}_{\mathrm{i}}} \boldsymbol{e}_{\mathrm{ij}} \boldsymbol{x}_{\mathrm{j}}^{\mathrm{l}-1}$$

where $\boldsymbol{x}_{\mathrm{i}}^{\mathrm{l}}$ is the node features of node i at the lth layer, $\boldsymbol{x}_{\mathrm{i}}^{0}$ is the ith row of the input node feature matrix F, and Θ is a learnable linear transformation, which maps the node features from shape [1,D] to [1, GCN-hidden]. Gi and eij are the neighbourhood of node i and the edge weight connecting nodes i and j given by the set of edges E respectively. The GCN-hidden is a tunable hyper-parameter of the GCN architecture. A rectified-linear-unit (ReLU) activation is applied to the output of GCN, and batch normalisation is performed. We refer to the node-wise outputs of GCN as node embeddings.

After L GCN layers are applied, the output is constructed by node embeddings in the form of a N×H matrix, where H is the hidden size given by GCN-hidden. In order to produce a graph-level embedding, a maximum readout layer is applied, resulting in an H-dimensional graph embedding r for each graph g .

$$\boldsymbol{r}_{\mathrm{g}} = \max_{\mathrm{i}=1}^{\mathrm{N}} \boldsymbol{x}_{\mathrm{i}}^{\mathrm{L}}$$

Module 5 : Experiments

In the experiment, the prediction effect was achieved by learning and classifying the EEG data characteristics of preictal and interictal segments. In order to evaluate the prediction effect of the prediction system, we introduce some performance evaluation indexes. In this paper, Accuracy, Sensitivity, Specificity, Precision, Recall, and F1-Score are used as the evaluation indexes for the model. The calculation formulae are as follows:

1. **Accuracy:**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision:**

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall:**

$$Recall = \frac{TP}{TP + FN}$$

4. **f1-score:**

$$f1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$f1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

where the true positive (TP) and true negative (TN) are correctly classified as preictal and interictal EEG segments, respectively. False positives (FP) and false negatives (FN) indicated that they were incorrectly predicted as preictal and interictal EEG segments.

AUC stands for the area under the curve of the receiver operating characteristic curve (ROC), which denotes the probability that the positive samples are assigned a higher score than the negative samples

Chapter 7: Software Testing

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual and organization. Projects which contain all documents have a high level of maturity. Careful documentation can save the time, efforts and wealth of the organization.

If the testing or development team gets software that is not working correctly and developed by someone else, so to find the error, the team will first need a document. Now, if the documents are available then the team will quickly find out the cause of the error by examining documentation. But, if the documents are not available then the tester need to do black box and white box testing again, which will waste the time and money of the organization. More than that, Lack of documentation becomes a problem for acceptance.

Benefits of using Documentation

- Documentation clarifies the quality of methods and objectives.
- It ensures internal coordination when a customer uses software application.
- It ensures clarity about the stability of tasks and performance.
- It provides feedback on preventive tasks.
- It provides feedback for your planning cycle.
- It creates objective evidence for the performance of the quality management system.

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the users point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as documentation testing.

This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

The most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the teams workload and keep track of the process. Lets take a look at the key requirements for these files and see how they contribute to the process.

- **Test strategy**
  An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

- **Test data**
  The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

- **Test plans**
  A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. Its the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

- **Test scenarios**
  In scenarios, testers break down the productÃ¢â‚¬â„¢s functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

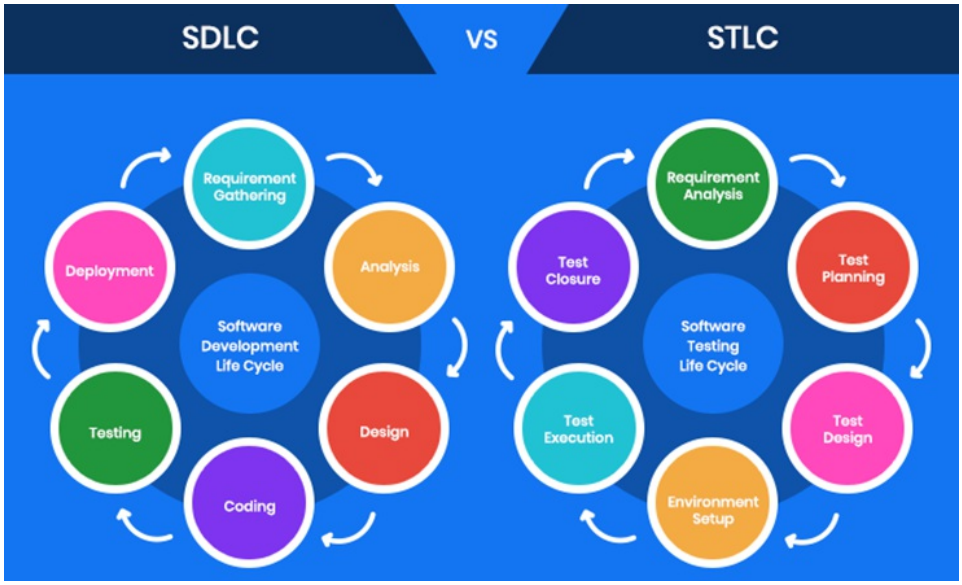
- **Test cases**
  If the test scenario describes the object of testing (what), a scenario describes a procedure (how). These files cover step-by-step guidance, detailed conditions, and current inputs of a testing task. Test cases have their own kinds that depend on the type of testing, functional, UI, physical, logical cases, etc. Test cases compare available resources and current conditions with desired outcomes and determine if the functionality can be released or not.

- **Traceability Matrix**
  This software testing documentation maps test cases and their requirements. All entries have their custom IDs Ã¢â‚¬â€ team members and stakeholders can track the progress of any tasks by simply entering its ID to the search.

The combination of internal and external documentation is the key to a deep understanding of all testing processes. Although stakeholders typically have access to the majority of documentation, they mostly work with external files, since they are more concise and tackle tangible issues and results. Internal files, on the other hand, are used by team members to optimize the testing process.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Software Testing Life Cycle (Taken from Google.com)



Introduction

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third-party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

- Java framework - JUnit
- PHP framework - PHPUnit
- C++ frameworks - UnitTest++ and Google C++
- .NET framework - NUnit
- Python framework - py.test

Software Testing Word Cloud (Taken from Google.com)

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements. For instance, seek answers to the following questions,

1. Are you able to login to a system after entering correct credentials?
2. Does your payment gateway prompt an error message when you enter incorrect card number?
3. Does your Add a customer screen adds a customer to your records successfully?

Test Driven Development

Test Driven Development, or TDD, is a code design technique where the programmer writes a test before any production code, and then writes the code that will make that test pass. The idea is that with a tiny bit of assurance from that initial test, the programmer can feel free to refactor and refactor some more to get the cleanest code they know how to write. The idea is simple, but like most simple things, the execution is hard. TDD requires a completely different mind set from what most people are used to and the tenacity to deal with a learning curve that may slow you down at first.

Functional Testing types include:

- Unit Testing
- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance Testing

Non-functional Testing types include

- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing
- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing

Unit Testing

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Most of us might know the classic definition of Unit Testing

Unit Testing is the method of verifying the smallest piece of testable code against its purpose

If the purpose or requirement failed then the unit test has failed. In simple words, Unit Testing means - writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

Blackbox Testing

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks appÃ¢â‚¬â„¢s functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

Integration Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

Component integration testing Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.

System Testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product. System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end systems actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customerÃ¢â‚¬â„¢s point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing;

Sanity Testing

Sanity Testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not. The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, the build is rejected by the testing team to save time and money. It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing.

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as Ã¢â‚¬oeDay 0Ã¢â‚¬ check. It is done at the build level.

It helps not to waste the testing time to simply testing the whole application when the key features donÃ¢â‚¬â„¢t work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

To conduct smoke testing, we do not write test cases. We just pick the necessary test cases from already written test cases. As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In

general, we pick minimal number of test cases that wont take more than half an hour to execute.

The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform.

Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure. After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing.

Regression Testing

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix. For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough varietyÃ¢â‚¬â€such as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prioritizationÃ¢â‚¬â€to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule. But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your companys bottom line.

Performance testing

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

Performance Testing (Taken from Google.com)

Performance testing gathers all the tests that verify an applications speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application. Performance testing is the testing that is performed to ascertain how the components of a system are performing under a certain given situation. Resource usage, scalability, and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of a software product.

Software Performance testing is type of testing perform to determine the performance of system to major the measure, validate or verify quality attributes of the system like responsiveness, Speed, Scalability, Stability under variety of load conditions. The system is tested under a mixture of load conditions and check the time required responding by the system under varying workloads. Software performance testing involves the testing of application under test to ensure that application is working as expected under variety of load conditions. The goal of performance testing is not only find the bugs in the system but also eliminate the

performance bottlenecks from the system.

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as Ã¢â‚¬oeEndurance testingÃ¢â‚¬ and Ã¢â‚¬oeVolume testingÃ¢â‚¬. The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load. Load testing comes under the Non Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is perform to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.v

Testing printer by sending large job. Editing a very large document for testing of word processor Continuously reading and writing data into hard disk. Running multiple applications simultaneously on server. Testing of mail server by accessing thousands of mailboxes In case of zero-volume testing & system fed with zero load.

Chapter 8: Sample Code

Chapter 9: Sample Output

Chapter 10: Conclusion

Our work introduces a novel neural network model architecture inspired by neurology, designed to comprehend brain activities across various temporal and spatial regions. The proposed comprises the following three primary components: (a) a node-feature-encoding-and-adjacency-matrices-construction block, specialized in capturing frequency-specific features based on an attention mechanism and generating task-specific adjacency matrices; (b) a graph aggregation block, dedicated to integrating task-specific information into the graph structure using an attention mechanism; and (c) a graph-feature-fusion-and-classification block, responsible for fusing features from diverse graph representations and executing the final classification.

Graph Neural Networks (GNNs) have shown significant promise in advancing brain connectivity analysis, enabling a more nuanced understanding of the human brain's complex network dynamics. By leveraging GNNs, researchers are now able to model intricate, non-linear relationships in brain networks, integrate multimodal imaging data, and explore dynamic connectivity patterns that are essential for capturing real-time brain activity. These capabilities represent an important shift from traditional graph-theoretical approaches, facilitating improved diagnostics, prognostics, and personalized therapeutic strategies for neurological disorders.

In this study, a classification model based on 2D representation of EEGs and a scalable neural network was proposed to improve the adaptability of the ADHD detection model to different hardware resource constraints and to improve the convenience of adopting new algorithms for different parts in the model.

Chapter 11: Future Work

In future work, some opportunity areas could be explored: First, a more extended analysis is required to evaluate the FSMs in combination with deep learning models, including a more extensive parameter tuning process and the use of more complex features. Furthermore, considering that our results showed that RF-ERF obtained one of the best performances, it would be interesting to perform an evaluation of tree-based models for feature selection and classification of ADHD and no-ADHD EEG data.

Chapter 10: Appendix - I Screenshots

Chapter 11: Appendix - II Sample Coding

Chapter 12: References

- Ying Chen,Yao Wang,Yibin Tang,Xiaojing Meng Predicting Disease Severity in Children With Attention Deficit Hyperactivity Disorder Using Dual-Branch Hypothesis Network IEEE Access, 2024
- Fujun Zhao,Chao Yang,Yi Zheng The N270 in Facial S1-S2 Paradigm as a Biomarker for Children With Attention-Deficit/Hyperactivity Disorder IEEE Access, 2020
- Zhihao Zhang,Simin Kang,Jifan Yu,Heyang Li,Gaohan Yin,Hongxing Zhang,Li Sun,Dangxiao Wang Quantitative Identification of ADHD Tendency in Children With Immersive Fingertip Force Control Tasks IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2023
- Shuaiqi Liu,Ling Zhao,Xu Wang,Qi Xin,Jie Zhao,David S. Guttery,Yu-Dong Zhang Deep Spatio-Temporal Representation and Ensemble Classification for Attention Deficit/Hyperactivity Disorder IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020
- Ree Nah Chua,Yuan Wen Hau,Chew Ming Tiew,Wan Leong Hau Investigation of Attention Deficit/Hyperactivity Disorder Assessment Using Electro Interstitial Scan Based on Chronoamperometry Technique IEEE Access, 2019
- Chun-Chuan Chen,Eric Hsiao-Kuang Wu,Yan-Qing Chen,Ho-Jung Tsai,Chia-Ru Chung,Shih-Ching Yeh Neuronal Correlates of Task Irrelevant Distractions Enhance the Detection of Attention Deficit/Hyperactivity Disorder IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2023
- Bilikis Banire,Dena Al-Thani,Marwa Qaraqe,Kamran Khowaja,Bilal Mansoor The Effects of Visual Stimuli on Attention in Children With Autism Spectrum Disorder: An Eye-Tracking Study IEEE Access, 2020
- Liang Zou,Jiannan Zheng,Chunyan Miao,Martin J. Mckeown,Z. Jane Wang 3D CNN Based Automatic Diagnosis of Attention Deficit Hyperactivity Disorder Using Functional and Structural MRI IEEE Access, 2017
- Yibin Tang,Xufei Li,Ying Chen,Yuan Zhong,Aimin Jiang,Chun Wang High-Accuracy Classification of Attention Deficit Hyperactivity Disorder With l2,1-Norm Linear Discriminant Analysis and Binary Hypothesis Testing IEEE Access, 2020
- Jungpil Shin,Sota Konnai,Md. Maniruzzaman,Md. Al Mehedi Hasan,Koki Hirooka,Akiko Megumi,Akira Yasumura Identifying ADHD for Children With Coexisting ASD From fNIRs Signals Using Deep Learning Approach IEEE Access, 2023
- Honorine Niyigena Ingabire,Haibo Qu,Min Li,Sixuan He,Joan Toluwani Amos,Yan Cui,Qing Wang,Dezhong Yao,Dan Ma,Peng Ren Stability Analysis of fMRI BOLD Signals for Disease Diagnosis IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2022
- G. Polanczyk and P. Jensen, â€œEpidemiologic considerations in attention deficit hyperactivity disorder: A review and update,â€ Child Adolescent Psychiatric Clinics North Amer., vol. 17, no. 2, pp. 245â€“260, Apr. 2008.
- Z. Zhang, G. Li, Y. Xu, and X. Tang, â€œApplication of artificial intelligence in the MRI classification task of human brain neurological and psychiatric diseases: A scoping review,â€ Diagnostics, vol. 11, no. 8, p. 1402, Aug. 2021.
- H. Chen, Y. Song, and X. Li, â€œA deep learning framework for identifying children with ADHD using an EEG-based brain network,â€ Neurocomputing, vol. 356, pp. 83â€“96, Sep. 2019.
- S. Su, J. Zhao, Y. Dai, L. Lin, Q. Zhou, Z. Yan, L. Qian, W. Cui, M. Liu, H. Zhang, Z. Yang, and Y. Chen, â€œAltered neurovascular coupling in the children with attention-deficit/hyperactivity disorder: A comprehensive fMRI analysis,â€ Eur. Child Adolescent Psychiatry, vol. 33, no. 4, pp. 1081â€“1091, Apr. 2024.
- L. Zou, J. Zheng, C. Miao, M. J. Mckeown, and Z. J. Wang, â€œ3D CNN based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural MRI,â€ IEEE Access, vol. 5, pp. 23626â€“23636, 2017.
- G. Polanczyk, M. S. de Lima, B. L. Horta, J. Biederman, and L. A. Rohde, â€œThe worldwide prevalence of ADHD: A systematic review and metaregression analysis,â€ Amer. J. Psychiatry, vol. 164, no., pp. 942â€“948, Jun. 2007.
- R. Thomas, S. Sanders, J. Doust, E. Beller, and P. Glasziou, â€œPrevalence of attention-Deficit/Hyperactivity disorder: A systematic review and meta-analysis,â€ Pediatrics, vol. 135, no. 4, pp. e994â€“e1001, Apr. 2015.
- T. Wang, K. Liu, Z. Li, Y. Xu, Y. Liu, W. Shi, and L. Chen, â€œPrevalence of attention deficit/hyperactivity disorder among children and adolescents in China: A systematic review and meta-analysis,â€ BMC Psychiatry, vol. 17, no. 1, p. 32, Dec. 2017.
- R. A. Barkley, â€œBehavioral inhibition, sustained attention, and executive functions: Constructing a unifying theory of ADHD,â€ Psychol. Bull., vol. 121, no. 1, pp. 65â€“94, 1997.
- M. A. Pievsky and R. E. McGrath, â€œThe neurocognitive profile of attention-Deficit/Hyperactivity disorder: A review of meta-analyses,â€

Arch. Clin. Neuropsychology, vol. 33, no. 2, pp. 143â€“157, Mar. 2018.

» G. V. Polanczyk, E. G. Willcutt, G. A. Salum, C. Kieling, and L. A. Rohde, â€œADHD prevalence estimates across three decades: An updated systematic review and meta-regression analysis,â€ Int. J. Epidemiol., vol. 43, no. 2, pp. 434â€“442, Apr. 2014.

» F. Li, â€œPrevalence of mental disorders in school children and adolescents in China: Diagnostic data from detailed clinical assessments of 17,524 individuals,â€ J. Child Psychol. Psychiatry, vol. 63, no. 1, pp. 34â€“46, Jan. 2022.

» J. Biederman, M. Fitzgerald, A.-M. Kirova, K. Y. Woodworth, I. Biederman, and S. V. Faraone, â€œFurther evidence of morbidity and dysfunction associated with subsyndromal ADHD in clinically referred children,â€ J. Clin. Psychiatry, vol. 79, no. 5, p. 20176, Aug. 2018.

» J. Posner, G. V. Polanczyk, and E. Sonuga-Barke, â€œAttention-deficit hyperactivity disorder,â€ Lancet, vol. 395, no. 10222, pp. 450â€“462, 2020.

» A. B. Fuermaier, â€œPerception in attention deficit hyperactivity disorder,â€ ADHD Attention Deficit Hyperactivity Disorders, vol. 10, no. 1, pp. 21â€“47, 2017.

» J. Wiklund, C. Lomberg, L. AlkÃ¦rsig, and D. Miller, â€œWhen ADHD helps and harms in entrepreneurship: An epidemiological approach,â€ Acad. Manage. Proc., vol. 2019, no. 1, p. 17481, 2019.

» J. C. Agnew-Blais, G. V. Polanczyk, A. Danese, J. Wertz, T. E. Moffitt, and L. Arseneault, â€œYoung adult mental health and functional outcomes among individuals with remitted, persistent and late-onset ADHD,â€ Brit. J. Psychiatry, vol. 213, no. 3, pp. 526â€“534, Jun. 2018.

» Z. Hawi, H. Yates, L. Kent, M. Gill, and M. Bellgrove, â€œA case-control genome wide association study of childhood attention deficit hyperactivity disorder (ADHD),â€ Eur. Neuropsychopharmacol., vol. 29, p. 956, Mar. 2019.

» K. Rubia, â€œFunctional connectivity changes associated with fMRI neurofeedback of right inferior frontal cortex in adolescents with ADHD,â€ NeuroImage, vol. 188, pp. 43â€“58, Mar. 2019.

» S. H. Hojjati, A. Ebrahimzadeh, A. Khazaee, and A. Babajani-Feremi, â€œPredicting conversion from MCI to AD by integrating rs-fMRI and structural MRI,â€ Comput. Biol. Med., vol. 102, pp. 30â€“39, Nov. 2018.

» Diagnostic and Statistical Manual of Mental Disorders, 5th ed., Amer. Psychiatric Assoc., Philadelphia, PA, USA, 2013.

» M. L. Danielson, R. H. Bitsko, R. M. Ghandour, J. R. Holbrook, M. D. Kogan, and S. J. Blumberg, â€œPrevalence of parent-reported ADHD diagnosis and associated treatment among US children and adolescents, 2016,â€ J. Clin. Child Adolescent Psychol., vol. 47, no., pp. 199â€“212, 2018.

» W. J. Barbaresi, R. C. Colligan, A. L. Weaver, R. G. Voigt, J. M. Killian, and S. K. Katusic, â€œMortality, ADHD, and psychosocial adversity in adults with childhood ADHD: A prospective study,â€ Pediatrics, vol. 131, no. 4, pp. 637â€“644, 2013.

» T. J. Spencer, J. Biederman, and E. Mick, â€œAttention-deficit/hyperactivity disorder: Diagnosis, lifespan, comorbidities, and neurobiology,â€ J. Pediatric Psychol., vol. 32, no. 6, pp. 631â€“642, 2007.

» S. BÃ¡lint, P. Czobor, S. KomlÃ³si, A. Meszaros, V. Simon, and I. Bitter, â€œAttention deficit hyperactivity disorder (ADHD): Gender- and age-related differences in neurocognition,â€ Psychol. Med., vol. 39, no. 8, pp. 1337â€“1345, 2009.

» S. R. Pliszka, â€œPatterns of psychiatric comorbidity with attention-deficit/hyperactivity disorder,â€ Child Adolescent Psychiatric Clinics North Amer., vol. 9, pp. 525â€“540, Jul. 2000.

» K. Konrad and S. B. Eickhoff, â€œIs the ADHD brain wired differently? A review on structural and functional connectivity in attention deficit hyperactivity disorder,â€ Hum. Brain Mapping, vol. 31, pp. 904â€“916, Jun. 2010.

» B. Abibullaev and J. An, â€œDecision support algorithm for diagnosis of ADHD using electroencephalograms,â€ J. Med. Syst., vol. 36, pp. 2675â€“2688, Aug. 2012.

» T. E. Elder, â€œThe importance of relative standards in ADHD diagnoses: Evidence based on exact birth dates,â€ J. Health Econ., vol. 29, pp. 641â€“656, Sep. 2010.

» G. Yong-Liang, P. Robaey, F. Karayanidis, M. Bourassa, G. Pelletier, and G. Geoffroy, â€œERPs and behavioral inhibition in a Go/No-go task in children with attention-deficit hyperactivity disorder,â€ Brain Cogn., vol. 43, pp. 215â€“220, Aug. 2000.

» Diagnostic and Statistical Manual of Mental Disorders (DSM-5), Amer. Psychiatric Assoc., Amer. Psychiatric Publishing, Washington, DC, USA, 2013.

» P. M. Dietz, C. E. Rose, D. McArthur, and M. Maenner, â€œNational and state estimates of adults with autism spectrum disorder,â€ J. Autism Develop. Disorders, vol. 50, pp. 1â€“9, May 2020.

» R. Brewer, F. HappÃ©, R. Cook, and G. Bird, â€œCommentary on â€˜autism, oxytocin and interoceptionâ€™: Alexithymia, not autism spectrum disorders, is the consequence of interoceptive failure,â€ Neurosci. Biobehav. Rev., vol. 56, pp. 348â€“353, Sep. 2015.

» J. L. Matson and K. R. M. Smith, â€œCurrent status of intensive behavioral interventions for young children with autism and PDD-NOS,â€

Res. Autism Spectr. Disorders, vol. 2, no. 1, pp. 60â€“74, Jan. 2008.

» T. Smith and S. Iadarola, â€œEvidence base update for autism spectrum disorder,â€ J. Clin. Child Adolescent Psychol., vol. 44, no. 6, pp. 897â€“922, Nov. 2015.

» G. V. Polanczyk, E. G. Willcutt, G. A. Salum, C. Kieling, and L. A. Rohde, â€œADHD prevalence estimates across three decades: An updated systematic review and meta-regression analysis,â€ Int. J. Epidemiol., vol. 43, no., pp. 434â€“442, 2014.

» A. Eloyan, â€œAutomated diagnoses of attention deficit hyperactive disorder using magnetic resonance imaging,â€ Frontiers Syst. Neurosci., vol. 6, p. 61, 2012.

» X. Peng, P. Lin, T. Zhang, and J. Wang, â€œExtreme learning machine-based classification of ADHD using brain structural MRI data,â€ PLoS ONE, vol. 8, no. 11, p. e79476, 2013.

» Y. Zang, â€œAltered baseline brain activity in children with ADHD revealed by resting-state functional MRI,â€ Brain Develop., vol. 29, no. 2, pp. 83â€“91, 2007.

» H. Yang, â€œAbnormal spontaneous brain activity in medication-naive ADHD children: A resting state fMRI study,â€ Neurosci. Lett., vol. 502, no. 2, pp. 89â€“93, 2011.

» G. Polanczyk and P. Jensen, â€œEpidemiologic considerations in attention deficit hyperactivity disorder: A review and update,â€ Child Adolescent Psychiatric Clinics North Amer., vol. 17, no., pp. 245â€“260, Apr. 2008.

» G. Polanczyk and P. Jensen, â€œEpidemiologic considerations in attention deficit hyperactivity disorder: A review and update,â€ Child Adolescent Psychiatric Clinics North Amer., vol. 17, no., pp. 245â€“260, Apr. 2008.