CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Software Quality Evaluation of AI/ML-Based Neuroimaging Tools

A Simulation Study Using BRAPH 2

Master's Thesis in Computer Science and Engineering

Yuxin Guo

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's Thesis 2025

Software Quality Evaluation of AI/ML-Based Neuroimaging Tools

A Simulation Study Using BRAPH 2

Yuxin Guo

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Software Quality Evaluation of AI/ML-Based Neuroimaging Tools
A Simulation Study Using BRAPH 2
Yuxin Guo

Supervisor: Giovanni Volpe, Yu-Wei Chang, Department of Physics, University of Gothenburg
Examiner: Gregory Gay, Department of Computer Science and Engineering, Chalmers and the University of Gothenburg

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Yuxin Guo
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The increasing integration of artificial intelligence (AI) and machine learning (ML) into neuroimaging research has amplified concerns regarding the quality, reproducibility, and trustworthiness of such software systems. Existing evaluation practices often lack standardized and systematic frameworks, limiting their ability to test software quality, especially for AI/ML-based neuroimaging software system. This study addresses this gap by developing and applying a simulation-based framework to evaluate key quality attributes—transparency, functional correctness, and robustness—in AI/ML-based brain imaging analysis pipelines. The framework leverages the Watts–Strogatz network model to generate controlled, simulated brain connectivity datasets, enabling rigorous and repeatable testing. Two analysis pipelines within BRAPH 2 (Brain Analysis using Graph Theory 2), an open-source MATLAB-based neuroimaging software for brain network analysis, are tested: a graph theory–based pipeline and a deep learning–based analysis pipeline. Both pipelines are evaluated against simulated datasets, successfully identifying the predefined salient brain regions and maintaining stable performance across repeated runs and random noise situations. Transparency was further enhanced through a graphical user interface and visualization modules that allow inspection of intermediate and final outputs. These results demonstrate that the proposed framework can effectively verify critical quality attributes in a controlled environment. The established methodology provides a robust and accessible foundation for extending validation to real-world neuroimaging datasets and for guiding future st andards in the quality evaluation of AI/ML-based neuroscience software.

Keywords: software quality, neuroimaging, AI/ML system, BRAPH 2, graph theory, deep learning, transparency, functional correctness, robustness

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Professor Giovanni Volpe for his invaluable guidance and trust. As a Chinese saying goes, "Genius is not scarce, but those who recognize it are." At the very beginning of my research journey in Sweden, Giovanni was that mentor who respected and believed in me. Our acquaintance began with nothing more than a single email, yet that message opened the door to a precious opportunity — to step into the world of scientific research and to enter a place where ideas collide and inspiration flourishes. Giovanni has not only provided me with academic guidance and support but has also shown me genuine respect and encouragement. His mentorship has inspired me to explore fearlessly and to innovate boldly. Under his help, I had the oppotunity to join the vibrant Softmatter Lab, where every group meeting and retreat was a source of great learning. Those experiences broadened my scientific horizon and deepened my appreciation for the boundless potential and beauty of AI4Science.

I am especially grateful to Yu-Wei Chang for his kind guidance and constant care during this period. Your guidance extended far beyond research itself — from the warm lunchboxes and thoughtful gifts to your genuine concern for my well-being, you brought warmth and strength to the long and cold Nordic winter. In supervising my thesis, you have done far more than what was required of you, and it is precisely your kindness and generosity that gave confidence and courage to a confused young man far away from home. Your wholehearted dedication to the BRAPH 2 software has also set an inspiring example for me. You have taught me that doing research requires not only perseverance and patience but also faith and passion for the future. I sincerely wish you the very best in your upcoming graduation and all success in the journey ahead.

I would also like to thank Xinwen Zhang, Jiacheng Huang, Fredrik Skärberg, Gan Wang, and Yan Chen for their help and encouragement. Your kindness has made my research life abroad feel much warmer and less lonely. On the top of that, thanks goes to Wenda Li, whose culinary talent and generosity always brought us together, adding a touch of joy and comfort to everyday life. These small yet meaningful moments reminded me that science is not only about data and experiments but also about people and the warmth of collaboration.

The past two years in Sweden have been filled with joy, challenges, regrets, and growth. Looking back, every attempt, every discussion, and every time I stood up after failure has become a precious part of my journey. As a Chinese proverb says, "A man is taller than the mountain, and his feet longer than the road." Now, as it comes time to say goodbye, I carry with me all these valuable experiences and memories. I hope that, wherever we go, we will continue to shine in our own ways — staying true to our original aspirations and continuing to explore the unknown with curiosity and courage.

Yuxin Guo, Beijing, November 2025

Glossary

AI (Artificial Intelligence): Refers to the development of computer systems that can perform tasks typically requiring human intelligence, such as learning, reasoning, problem-solving, and perception.

ML (Machine Learning): A subfield of AI that focuses on algorithms and statistical models which enable systems to learn from data and improve performance over time without being explicitly programmed.

GUI (Graphical User Interface): A user interface that allows interaction with electronic devices through graphical elements like buttons, icons, and windows, rather than text-based commands.

ISO/IEC TS 25058:2024: A technical specification published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), providing standardized quality evaluation guidelines specifically tailored for AI/ML-based software systems.

ROIs (Regions of Interest): In neuroimaging, ROIs refer to specific anatomical areas of the brain that are selected for focused analysis, often because they are hypothesized to be functionally or structurally significant.

BRAPH 2 (Brain Analysis using Graph Theory 2): BRAPH 2 is a MATLAB-based open-source software platform designed for brain network analysis using graph theory.

MRI (Magnetic Resonance Imaging): MRI is a medical imaging technique that uses magnetic fields and radio waves to create detailed images of the brain's structure.

ADHD (Attention-Deficit/Hyperactivity Disorder): A neurodevelopmental disorder characterized by persistent patterns of inattention, hyperactivity, and impulsivity that interfere with functioning or development. Key features include difficulty sustaining attention, excessive activity or restlessness, impulsive actions, early onset (before age 12), and symptoms present in more than one setting (e.g., home, school, work).

fMRI (Functional Magnetic Resonance Imaging): fMRI measures brain ac-

tivity by detecting changes in blood oxygen levels, allowing researchers to observe functional processes in the brain.

MLP (Multilayer Perceptron): A type of feedforward artificial neural network consisting of an input layer, one or more hidden layers, and an output layer. It is commonly used to model complex nonlinear relationships in data. Key parameters include the number of hidden layers, number of neurons per layer, activation functions, learning rate, batch size, optimizer/solver, and number of training epochs.

ABC (Actors, Behavior, and Context): A research framework in software engineering aiming for three desirable properties in empirical studies: generalizability over Actors (A), precise measurement of Behavior (B), and realism of Context (C).

DTI (Diffusion Tensor Imaging): An MRI technique that measures the diffusion of water molecules in biological tissue to assess white matter microstructure. Key features include measurements of diffusivity magnitude (e.g. mean diffusivity), directionality of diffusion (anisotropy, such as fractional anisotropy), and the ability to estimate neural fiber orientations for tractography.

EEG (Electroencephalography): A non-invasive technique for recording the brain's electrical activity via electrodes placed on the scalp. Key features include high temporal resolution; measurement of voltage fluctuations (brain waves) that reflect neural activity (especially from cortical neurons), and use in diagnosing and monitoring neurological functions and disorders.

ECG (Electrocardiography): A non-invasive medical test that records the heart's electrical activity from electrodes placed on the surface of the body. Key features include measurement of heart rate and rhythm, detection of impulses' origin and conduction pathways, and visualization of characteristic waveforms such as P wave, QRS complex, and T wave.

Contents

List of Figures

List of Tables

1

Introduction

Software has become essential for advancing our understanding of the brain in contemporary neuroscience research [1]. The complexity and diversity of brain imaging analysis require robust software frameworks to ensure scientific integrity through reliable software quality. This need is particularly critical when handling brain imaging data, as such data inherently exhibit more significant variability and complexity [2], [3] as research develops. In other words, the emergence of complex neuroimaging datasets has created an urgent demand for tools that are reliable, flexible, and reproducible [4].

While AI shows promising applications in neuroscience through network-based models for analyzing brain connectivity patterns, quality assurance in this field still mainly relies on evaluations through empirical data analyses driven by intuition and experience, leading to questions about the algorithm accuracy and reproducibility of the calculation [5], [6], [7]. For instance, research studies often validate software-generated results by comparing them to previous findings in the literature. However, these earlier studies frequently lack detailed information about the analysis setup, such as the rationale behind selecting specific ROIs (Regions of Interest), which is often unclear [8], [9]. This subjective approach potentially introduces bias and reduces reproducibility, highlighting the need for more standardized, data-driven evaluation frameworks.

Currently, several AI/ML-based neuroscience software tools exist. For example, BRAPH 2 has been successfully applied in neurosciences, providing deep learning analysis for brain imaging analysis [10]. However, many medical researchers using these AI/ML-based software tools do not have a background in computer science or a detailed understanding of how such systems work, particularly in software engineering and AI/ML algorithms. This creates a need for AI/ML-based software that can seamlessly integrate advanced AI/ML algorithms while also automating fundamental but standardized tasks such as quality check of AI/ML-based software system. By doing so, researchers can focus on generating scientific insights rather than dealing with technical challenges.

Therefore, while various neuroscience software tools have been successfully applied in different research contexts, the comprehensive evaluation of AI/ML-based software quality from a software engineering perspective remains underexplored. Yet, assess-

ing software quality is crucial: when analytical tools lack transparency, their processing steps and decision logic become opaque, obscuring potential implementation errors; when they lack functional correctness, undetected algorithmic or modeling biases may distort results; and when they lack robustness, unstable computations under varying data conditions can yield inconsistent or unreliable outcomes, ultimately leading to misleading scientific conclusions and reduced trust in neuroimaging findings. This study aims to fill this gap by employing experimental simulations to assess the quality of AI/ML-based software in neuroscience, using BRAPH 2 as a case study. By systematically evaluating software quality in accordance with ISO standards from the software engineering perspective, this research seeks to improve the overall reliability of AI/ML neuroscience software systems in the field. Through careful assessment of defined quality metrics, it establishes a benchmark for evaluating neuroimaging software and promotes higher standards for future development in this rapidly evolving domain.

1.1 Research Questions

Currently, AI/ML-based neuroscience software lacks a thorough and systematic discussion regarding quality control and evaluation, particularly in relation to established standards from software engineering. This gap creates significant uncertainty about the reliability and trustworthiness of such tools. Many studies [7] rely heavily on real-world data and expert intuition for algorithm parameter selection, making the analysis less reproducible, harder to generalise, and often inefficient. Without a simulation-based environment, researchers cannot precisely control data variability or evaluate software performance when the ground truth of brain connectivity is unknown, and input variability (e.g., scanner noise, preprocessing differences, demographic diversity) cannot be systematically controlled [11].

This challenge becomes even more complex when traditional neuroscience methods are combined with AI/ML techniques, particularly those involving both graph theory and deep learning. In such cases, it is often unclear which software quality attributes should be evaluated. Existing studies tend to emphasise reproducibility while overlooking other critical aspects such as functional correctness, traceability, and transparency. As a result, current approaches often fall short of delivering robust and reliable analysis pipelines.

To address this, the present study systematically explores how software quality can be assessed through experimental simulations—that is, by precisely controlling and testing algorithm behaviour under known conditions.

Research Question:
How can quality attributes of AI/ML-based neuroscience software be effectively evaluated to ensure reliable performance in real-world neuroimaging analysis pipelines?

To explore this question, this study uses BRAPH 2 as a case study, focusing on

a neuroimaging pipeline designed to identify group differences using both graph theory-based and deep learning-based methods. A central research goal in such pipelines is to determine which brain regions contribute most to the group differences (e.g., distinguishing patients from healthy controls). Brain regions are modelled as nodes in a graph, with connections representing interactions between them.

To simulate the brain network, we generate data in which a known set of brain regions carries the most discriminative information. The hypothesis is to test whether the pipeline can correctly identify these regions by establishing a controlled simulation environment. This study will provide insight into its transparency, functional correctness, and robustness. A detailed literature review, methodology, and results will be described in the following chapters.

1.2 Aim

This study aims to systematically analyzes, identify, and evaluate critical quality attributes required for ensuring AI/ML-based software systems in brain neuroscience. Leveraging the BRAPH 2 framework, the research will specifically address challenges in the quality of AI/ML-based neuroscience software systems. The purposes are as follows:

1. To design and implement an evaluation framework that brings together graph theory-based and deep learning-based analysis approaches, allowing for flexible parameter manipulation and systematic quality assessment within a controlled environment, thereby facilitating rigorous verification of software attributes on both simulated and real-world brain imaging data.

2. To create a practical and easy-to-use testing approach that supports researchers without software engineering expertise in independently evaluating critical quality aspects of neuroscience software, including transparency, functional reliability, and robustness under varying conditions.

Through these objectives, this research aims to significantly enhance the robustness and transparency of AI/ML-based neuroscientific software, providing tools that allow precise algorithm control and increased reliability in neuroscientific research.

1.3 Significance of the Study

The significance of this study lies in its potential to set novel and practical standards for evaluating AI/ML-based software quality within the neuroscientific community. By systematically exploring and validating critical quality attributes using the BRAPH 2 software, this study will fill a current void in the comprehensive software quality evaluation of neuroscience software that utilizes AI/ML methods.

First, it provides neuroscientists with accessible methodologies for robust software evaluation, which is essential for advancing scientific integrity in the face of increas-

ingly complex AI/ML-based analyses. Secondly, it demonstrates how a systematic and user-friendly framework can support precise control over analysis parameters and enable rigorous evaluation of neuroscience pipelines, reducing reliance on manual adjustments and subjective experience-based controls. Lastly, in collaboration with the Karolinska Institute and using real datasets to verify its actual application effect, the insights and methodologies established through this research will offer valuable benchmarks and standards for evaluating future neuroscience software, thereby fostering broader application and development across the discipline.

In summary, this work serves both as a practical guide for neuroscience software developers and as a foundational reference for ongoing research in AI/ML-based neuroscience applications.

1.4 Report Outline

This thesis is structured into six main chapters, each serving a distinct purpose in addressing the research question on software quality evaluation in AI/ML-based neuroimaging software.

Chapter 1: Introduction outlines the research background, motivation, and research questions, clarifies the goals and contributions, identifies gaps in transparency, functional correctness, and robustness in current AI/ML neuroimaging software evaluation, and introduces BRAPH 2 as the case vehicle together with a simulation-based validation approach.

Chapter 2: Background reviews three foundations: software quality (and relevant standards) in AI/ML systems, the ecosystem of neuroimaging analysis software, and related research and practice. Drawing on the literature and application scenarios, it motivates transparency, functional correctness, and robustness as the key quality attributes prioritized in this work.

Chapter 3: Methods presents the study design and methodology, including a controlled, simulation-based testing environment (e.g., using Watts–Strogatz model to generate data with controllable connectivity patterns), the design of two analysis pipelines verification in BRAPH 2 (graph theory and deep learning), and the experimental schemes and the metrics used to examine each quality attribute (transparency, functional correctness and robustness).

Chapter 4: Results reports the validation of the quality-evaluation framework. It focuses on the verification of the graph theory-based and deep learning-based analysis pipeline, explains, and summarizes the testing results. Evidence and observations are presented from the perspectives of transparency, functional correctness, and robustness.

Chapter 5: Discussion interprets the results in relation to the research question, addressing the framework's applicability and boundaries for neuroimaging software quality evaluation. It discusses implications for developers and end-user researchers,

the generalizability and limitations of the approach, threats to validity, and relevant ethics and guidance for the use of generative AI.

Chapter 6: Conclusion and Future Work summarizes the main findings and contributions, highlighting the value and practice of simulation-driven quality evaluation for AI/ML neuroimaging software. It also outlines next steps, including completing deep-learning pipeline tests, external validation on real-world datasets (e.g., collaborations with clinical/research institutions), and extending the framework toward a more generalizable quality-assessment toolkit.

2

Background

This chapter provides the foundational background for understanding the intersection between AI/ML and software quality, particularly in the context of neuroscience applications. It begins by outlining the key principles and challenges of ensuring software quality in AI/ML systems. It then explores how these principles apply within the field of neuroscience, where the increasing use of AI has introduced both new opportunities and new complexities. The chapter also discusses the specific quality requirements of AI/ML-based neuroscience tools, focusing on attributes such as transparency, functional correctness, and robustness. Finally, it highlights existing gaps in current practice and research, motivating this project to systematically address these deficiencies by proposing and applying a structured framework for evaluating the quality of neuroscience software systems.

2.1 Software Quality in AI/ML-based Systems

Software quality is one of the core research directions in the field of software engineering, and its goal is to measure the key criteria of the comprehensive performance of software products in terms of meeting user needs, business objectives, and development and maintenance. Software quality not only affects the current performance of a software system but also determines its sustainable evolution and the ease of maintenance. In large-scale industrial software systems, well-known quality models and software metrics are often used to measure and improve software quality, and these models help deliver software products that meet or exceed customer expectations [12]. Companies like Alibaba have built systematic quality assessment platforms to achieve continuous quality monitoring and improvement [13]. In software systems for academic research, on the other hand, where the main goal is to advance scientific discovery and ensure the accuracy, stability, and reproducibility of its results. As we can see, software quality encompasses a number of dimensions, which may vary in different application domains, but generally encompasses multiple dimensions such as functionality, reliability, usability, efficiency, maintainability, and portability [14].

In the AI/ML software field, the quality problem is more complicated. On the one hand, the non-determinism of AI/ML models itself introduces new quality risks, and on the other hand, traditional software testing and evaluation methods are difficult

to adapt to the dynamic characteristics of deep learning systems [15]. Specifically, this challenge is exacerbated by the dependence on data features, the stochastic nature of inputs and outputs, and the unpredictability of decision-making due to the inherent cross-disciplinary uncertainty and probabilistic nature of AI/ML software systems. As AI/ML software is increasingly integrated into various domains, ensuring its reliability and quality has become a key area of research and interest in both academia and industry [16].

At the same time, the question of how to assess the quality of AI/ML software, and what quality attributes are critical for AI/ML software is also a question worth exploring. ISO/IEC TS 25058:2024 [17], a technical specification for software for AI/ML systems, refines the quality attributes of AI/ML-oriented software and provides comprehensive guidelines for quality assessment, with eight core quality attributes defined, including compatibility, reliability, usability, performance efficiency, portability, security, maintainability, and functional suitability. Together, these attributes form a comprehensive framework for assessing the quality of software for AI/ML systems, which not only covers the non-functional requirements of traditional software systems, but also reflects the unique challenges of AI/ML systems in terms of data, models, and prediction, etc., which are particularly important in AI/ML-based software systems, especially in high-uncertainty scenarios such as healthcare and scientific research [15].

Together, these attributes form a comprehensive framework for assessing the quality of software for AI/ML software systems, which not only covers the non-functional requirements of traditional software systems, but also reflects the unique challenges of AI systems in terms of data, models, and prediction, etc., which are particularly important in AI/ML-based software systems, especially in high-uncertainty scenarios such as healthcare and scientific research.

In the following subsections, they will examine how these quality attributes apply to AI/ML software systems, focusing on the specific challenges and evaluation needs within the neuroscience context.

2.2 Software in Neuroscience

With the deepening of neuroscience research, especially in brain imaging and neural network analysis, researchers have begun to rely on various software tools to process and analyze massive amounts of neuroimaging data. As the volume and complexity of neuroimaging data continue to grow, reliable software systems have become indispensable for ensuring accuracy, reproducibility, and scientific rigor. Many tools now support not only basic processing functions but also sophisticated modeling methods such as graph theory and deep learning, enabling a deeper understanding of brain function and pathology. These tools vary in their design, functionality, and degree of automation, depending on their target users and intended applications.

A wide range of software tools is employed in the preprocessing stage to prepare

Table 2.1: Summary of Key Neuroscience Software Tools

Software	Developer	Primary Functions	Citation Number*
FreeSurfer [18]	Harvard University's Martinos Center (2012)	MRI and PET (Positron Emission Tomography) Imaging Preprocessing	9320
SPM [19]	Wellcome Trust Centre for Neuroimaging, UCL (2011)	MRI and fMRI Imaging Preprocessing and Their Statistical Analysis	5534
FSL [20]	University of Oxford's FMRIB Centre (2012)	MRI, fMRI and Diffusion Brain Imaging Preprocessing	11781
NBS [21] & **NBS-Predict**[22]	Katherine L. Bottenhorn et al. (2024)	Brain Neuroimaging Connectome Analysis	2769
BRAPH 1 [23] & **2** [24]	Soft Matter Lab, University of Gothenburg (2017) & (2025)	Brain Neuroimaging Connectome Analysis	259

*Citation numbers were retrieved as of April 24, 2025.

neuroimaging data for further analysis. For example, FreeSurfer [18] specializes in cortical surface reconstruction and volumetric segmentation, offering robust tools for anatomical mapping using MRI (Magnetic Resonance Imaging). Similarly, SPM [19] is widely used for statistical analysis of functional imaging data such as fMRI (Functional Magnetic Resonance Imaging) and PET (Positron Emission Tomography), and is especially strong in hypothesis-driven task-based studies. Another versatile suite, FSL [20], covers a broader spectrum, including both structural and functional analyses, with capabilities for preprocessing, statistical modeling, and visualization. These tools typically handle steps like denoising, co-registration, normalization, and segmentation.

For data analysis, particularly at the network level, more specialized tools are available. NBS [21] extends the network-based statistical framework, a standard method for finding differences in brain networks by testing hypotheses about connected groups, with graph theory to support brain network-based predictive modeling. It emphasizes the interpretability of network substructures and supports cross-validation techniques, specifically, the procedure of repeatedly partitioning the available data into training and test (or validation) sets, building the predictive model on the training portion, and evaluating it on the held-out data, in order to estimate the model's ability to generalise to unseen individuals. BRAPH 2 [24], on the other hand, is tailored for advanced brain network analysis (e.g., multiplex graph theory analysis) and provides extensive support for both graph theory analysis and deep learning techniques. It is not just traditional graph measures and classification. Notably, deep learning has become increasingly integrated into these tools to im-

prove performance, automate workflows, and enhance interpretability, particularly in structural and functional brain network analysis.

Despite these advances in neuroscience software capabilities, relatively little attention has been paid to software quality, particularly for tools incorporating AI/ML. In particular, the complexity and uncertainty of scientific research software itself, as well as the lack of software engineering knowledge among developers., lead to more complex and unique challenges in assessing software quality [25]. Besides, as deep learning becomes increasingly integrated into neuroscience analysis pipelines, the need for robust quality assessment frameworks grows more critical. Traditional preprocessing steps now frequently incorporate neural network-based segmentation and classification, while analysis tools increasingly use machine learning for pattern recognition and prediction. However, the complex and probabilistic nature of these AI/ML methods introduces new challenges for software quality assurance that conventional testing approaches cannot adequately address. The next section will examine how software quality is currently approached in neuroscience tools and highlight the significant gap in quality assessment methodology for AI/ML-integrated neuroscience software, which forms the central motivation for this project.

2.3 Related Work

Medical AI/ML research has made tremendous progress, but its practical application in clinical practice remains limited. This is mainly because existing medical AI/ML softwares may have technical errors, biases, lack transparency, difficulty in interpretability, and data privacy and security risks [26]. These issues lead to insufficient trust in AI/ML tools among multistakeholders, including medical institutions, doctors, patients, and regulatory bodies. The FUTURE-AI [26] consensus guidelines, developed by an international coalition of 117 experts from 50 countries, identify six core principles for AI/ML tools as the Figure 2.1. To be specific, these principles include: Fairness, which ensures consistent performance across diverse populations and mitigates algorithmic bias; universality, promoting generalizability across different clinical settings and institutions; traceability, which emphasizes thorough documentation throughout the AI/ML lifecycle for accountability and regulatory compliance; usability, ensuring that tools address real clinical needs and are accessible to both healthcare professionals and patients; robustness, which requires AI/ML systems to maintain stable and reliable performance under varying or unforeseen data conditions; and explainability, which mandates transparent and interpretable decision-making processes to foster user trust. Collectively, these principles establish a foundation for the responsible and effective integration of AI into healthcare.

From a computer science perspective, the ISO/IEC TS 25058:2024 [17] technical standard provides an important framework for AI/ML software systems, particularly in neuroscience applications. It covers eight core quality parent attributes along with 32 subattributes that play a key role in the design, development, and evaluation of neuroscience software.

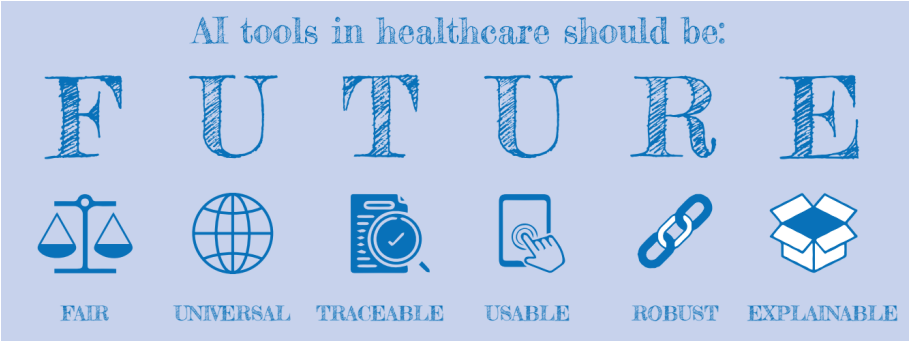

Figure 2.1: Organisational structure of the FUTURE-AI framework for trustworthy AI according to six guiding principles [26]. The framework is structured around six guiding principles—Fairness, Universality, Traceability, Usability, Robustness, and Explainability—which together form the acronym FUTURE. These principles were established by the FUTURE-AI Consortium through international consensus, bringing together 117 experts in artificial intelligence, medicine, ethics, and law from 50 countries. The original 55 requirements were distilled into six guiding principles and 30 best-practice recommendations, covering the full lifecycle of AI in healthcare—design, development, validation, deployment/monitoring, and governance—to guide the development and evaluation of trustworthy and deployable medical AI systems [26]. In the context of this thesis, these principles provide a systematic reference from the perspective of practical healthcare applications for evaluating quality attributes in AI/ML software workflows for neuroimaging analysis.

It can be seen that the FUTURE-AI [26] guidance framework and ISO/IEC TS 25058:2024 [17] technical standard overlap in many aspects of their content. However, not all of these attributes will play a decisive role. In the context of neuroscience software, identifying which attributes are most crucial requires an application-specific perspective grounded in empirical findings.

While no systematic software engineering evaluation has yet been conducted for most AI/ML-based neuroscience tools, and a formal taxonomy of prioritized attributes is still lacking, a review of existing research literature and case studies reveals that transparency, functional correctness, and robustness are consistently emphasized across studies addressing the quality evaluation of AI/ML-based neuroimaging tools from the software perspective.

In neuroscience applications, the importance of transparency, functional correctness, and robustness stems from the high-risk nature of medical AI/ML. Transparency ensures clinicians understand how decisions are made; functional correctness guarantees reliable performance in critical clinical scenarios; robustness ensures AI/ML maintains accuracy when facing common input data variations in neuroimaging and measurements.

The selection of these attributes is not arbitrary but driven by the unique challenges and high-stakes nature of neuroscience applications. The following sections will examine these three attributes in depth, using case studies and practical requirements to demonstrate why these three attributes are fundamental to ensuring the quality and trustworthiness of AI/ML-based neuroscience software, providing a focused framework for evaluation and development of such systems.

2.3.1 Transparency

Transparency refers to the ability of software systems to enable users to understand their operational processes, decision logic, and internal mechanisms [17]. In AI/ML-based systems, transparency is particularly important because complex algorithms and models (such as deep learning networks) are often viewed as "black boxes," meaning users cannot understand how the model makes predictions or decisions [27]. This necessitates clearly demonstrating model training, inference processes, and decision bases, so that users, developers, and auditors can understand how the model works, thereby enhancing software trustworthiness.

Since neuroimaging data analysis often involves high-risk tasks such as medical diagnosis, transparency can help doctors, researchers, and developers understand the model's decision-making process. For example, in clinical applications, the black box nature of AI/ML software may make it difficult for doctors to trust its recommended diagnoses, thus affecting their decision-making. Phan et al. [28] emphasize that in neuroimaging analysis, explainable AI/ML methods are crucial for explaining how models identify pathological features, especially when involving complex deep learning architectures. Meanwhile, research by Wang et al. [29] shows that a lack of transparency may lead to critical diagnostic errors, such as missed stroke diagnoses,

affecting patient outcomes, particularly in time-sensitive scenarios.

Brain imaging AI/ML must explain decisions to clinicians to build trust and meet regulatory requirements, especially in FDA-approved tools, ensuring accountability and traceability [30]. Research by Li et al. [31] highlights the crucial role of transparency in neuroimaging diagnostics, emphasizing the value of explainable AI techniques (such as heat maps) in improving clinicians' understanding and acceptance of system outputs. Bowring et al. [32] demonstrate that the lack of transparency can lead to significantly different results from various analysis software (such as AFNI, FSL, and SPM), which may have major impacts on the interpretation of research findings. Their study shows that analyzing the same dataset with different software can result in Dice coefficients as low as 0.000-0.743, highlighting the necessity of transparent documentation of analysis workflows and software configurations.

Furthermore, for neuroscience software, and any lack of transparency could lead to ethical issues. Zhang et al. [33] found that certain AI systems perform worse for patients with darker skin when diagnosing melanoma due to unrepresentative training data. The article suggests improving transparency through explainable AI (XAI) while emphasizing that ethical frameworks should be guided by bioethical principles such as autonomy, non-maleficence, beneficence, and justice. This is especially important for neuroscience software, as AI may influence decision-making in neurosurgery, and transparency ensures patients can provide informed consent and avoid potentially discriminatory outcomes. Additionally, Kiseleva et al. [34] mention that the European Union's General Data Protection Regulation (GDPR) requires AI explainability, which is particularly critical in neuroscience software because patient data is highly sensitive and any lack of transparency could lead to ethical issues.

2.3.2 Functional Correctness

Functional correctness refers to the ability of a system or software to accurately perform its functions according to expected specifications and requirements under any conditions, meaning the output results always conform to expected or defined correctness standards [17]. It encompasses the comprehensiveness, accuracy, and timeliness of system functions. For neuroscience software, functional correctness means that the software can provide accurate data processing, analysis, and output, and can perform under predetermined time and conditions.

Functional correctness includes the integrity and consistency of software; software errors may lead to serious clinical consequences, and functional correctness ensures that AI/ML software system outputs align with medical expectations, thereby ensuring patient safety. This requires that in neuroscience software, all expected functions work properly, and collaboration between different modules is seamless. For example, electroencephalogram (EEG) analysis software needs to accurately identify and analyze brain waves, ensuring that signal processing, feature extraction, and classification proceed as expected. If an error occurs at any stage, the results of the entire analysis process will be affected, potentially leading to incorrect clinical diagnoses or research conclusions. Research by Prasad et al. [35] shows that different versions

of FreeSurfer have significant differences in measuring indicators such as cortical thickness (ICC values as low as 0.37–0.61), indicating that even the same tool may exhibit functional correctness deviations across different versions. Additionally, research by Clerx et al. [36] different versions of FreeSurfer may introduce risks of misdiagnosis, thereby affecting treatment planning for neurological diseases, while also demonstrating that functional correctness can be gradually improved through tool iteration. The paper by Chatelain et al. [37] introduces a results stability testing method based on numerical variability, demonstrating through random perturbation experiments on fMRIPrep that even minor code or environmental changes can significantly alter outputs, highlighting the necessity of system function verification mechanisms.

Moreover, with the dramatic increase in quantity and rich variety of neuroimaging data, how to efficiently manage and apply these data, ensuring correctness under different conditions for clinical reliability across patient populations, has become an urgent problem in the field of neuroscience [38]. The brain imaging data analysis system developed by Zhang et al. [39] based on Docker containerization technology verifies system reliability and stability through standardized testing tools, highlighting the importance of functional correctness when processing clinical data. Lee et al.'s research [40] emphasizes the importance of maintaining functional correctness in neuroimaging analysis, especially when processing data from different sources and of varying quality. Fischl's research [41] shows that differences in data from different scanners or patient age groups may lead to variations in analysis results. Therefore, functional correctness is a core requirement for neuroscience software, ensuring the reliability and accuracy of analysis results, which in turn affects research conclusions and clinical decisions.

2.3.3 Robustness

Robustness represents the stability and fault tolerance of software when facing different types of inputs, missing data, or abnormal situations [17]. AI/ML software systems, especially AI models in neuroscience applications, must maintain efficiency and stability when facing complex data in the real world. In clinical environments, medical AI/ML software tools are affected by various data changes in actual application environments, as data often becomes unstable due to equipment differences, operator skill differences, data loss, and other factors. A system with insufficient robustness may fail in specific contexts, leading to misdiagnosis or missed diagnosis.

In neuroscience software, this means the system should be able to handle various types of data inputs, including incomplete data, noise interference, and outliers, to ensure real-world generalizability. For example, fMRI data analysis software should be able to provide reliable analysis results even in the presence of image artifacts caused by subject head movement. Research by Guo et al. [11] points out that neuroscience research relies on reproducible results, and robustness ensures that tools like SPM perform consistently across different studies, reducing inter-study variation. Abdelaziz et al. [42] emphasize that in neuroimaging analysis, models need to maintain robustness against different types of scanners, parameter settings,

and image quality variations to ensure reliability in clinical applications.

At the same time, robustness also means reducing cross-environment and cross-institutional adaptation capability [43]. Research by Plis et al. [43] indicates that in brain connectome analysis, robustness is a key indicator for evaluating model quality, as it directly affects the reliability and reproducibility of analysis results. Furthermore, Plis et al. emphasize that in neuroscience research, due to various variables in artificial data collection and processing, robustness is crucial for ensuring the reliability of research findings. Glatard et al.'s [44] research shows that even changes in operating systems can lead to differences in neuroimaging analysis results, highlighting the need for neuroscience software to maintain consistency across different computing environments. The multisite test study on FreeSurfer software [35] demonstrates that different versions of FreeSurfer exhibit varying failure rates in processing brain regions for quality control subsets, for example: "The left lingual, left cuneus, right parahippocampal, and right bank of the superior temporal sulcus all had considerable quality issues." This emphasizes the robustness that neuroscience software needs when processing real-world data. Lack of system robustness checks may lead to failure in real-world deployment, especially in time-sensitive scenarios.

2.4 Gap of Neuroscience Software Quality

Despite the growing application of AI/ML software in neuroscience, particularly in neuroimaging analysis, a significant gap remains in the systematic evaluation of software quality within this domain. As outlined in Section 2.1, software quality is a foundational aspect of software engineering, encompassing many critical attributes. However, most existing neuroscience software tools have not been developed with these software engineering principles in mind. In contrast to industrial AI/ML software systems, where structured quality frameworks are routinely applied, AI/ML-based neuroscience software often lacks standardized development and evaluation practices. This is particularly concerning given the high-stakes nature of clinical and scientific decision-making supported by such tools.

Section 2.2 further highlights that widely-used neuroscience tools such as FreeSurfer, SPM, FSL, NBS, and BRAPH 2 have either limited or no publicly documented efforts toward comprehensive software quality validation. Although these tools are powerful and widely accepted in the community, their quality assurance has typically focused on empirical performance with real-world data rather than on rigorous quality metrics or standardized testing protocols. This gap is even more pronounced in AI/ML-based systems like NBS and BRAPH 2, which introduce additional complexity due to their reliance on probabilistic models and data-driven algorithms.

As discussed in Section 2.3, the literature does point to specific quality concerns that repeatedly emerge in neuroscience AI/ML applications—namely, the lack of transparency, functional correctness, and robustness. These attributes are frequently cited in studies as critical for ensuring the reliability, safety, and trustworthiness of medical AI/ML software. However, these concerns have not been systematically ad-

dressed from a software engineering perspective. There are currently no established frameworks tailored to evaluating these specific attributes in the context of neuroscience software, nor are there domain-specific benchmarks or validation protocols in place.

Therefore, the gap in neuroscience software quality can be summarized from the software engineering perspective as follows:

Insufficient Quality Assessment in AI/ML-Based Neuroscience Software Systems

As AI/ML algorithms become increasingly integrated with neuroscience software, AI/ML-based neuroscience software has become the future research direction. However, as shown in the research content just enumerated, while they have raised various issues facing neuroscience software, the quality assessment of software incorporating AI/ML algorithms is still in its infancy, with relevant research and practice remaining very limited. The introduction of AI/ML algorithms has made software quality assessment more complex, particularly in aspects such as functional correctness, transparency, and robustness. This is because such research software is typically developed by researchers from non-computing fields whose primary goal is to advance scientific discovery rather than the engineering quality of the software itself. This leads to many research projects lacking standardized design documentation, quality control processes, and systematic testing frameworks during the development phase. Testing activities are often viewed as an additional burden, lacking a cultural foundation for systematic execution [25].

Challenges in Quality Assessment due to AI/ML Characteristics' Uncertainty

Due to the "black box" nature of AI/ML software system, data dependency, and algorithmic uncertainty [27], additional challenges are posed for software quality validation. In neuroimaging analysis, the decision-making process of AI/ML algorithms is often difficult to explain, and different inputs may lead to different outputs. This "unknownness" makes traditional software testing techniques, such as regression testing based on known expected results, difficult to apply directly. Furthermore, since the results themselves are the subject of research, it is difficult to judge the correctness of program outputs, which has become one of the core difficulties in research software quality assessment [45]. Therefore, how to verify the functional correctness and robustness of AI/ML algorithms, and ensure that even non-professional users can understand and verify their transparency has become a core issue in neuroscience software quality assessment.

In summary, neuroscience software quality assessment faces multiple challenges, particularly evident in the new field where AI/ML and neuroscience converge. Traditional software quality assessment frameworks struggle to address the "black box" nature of AI/ML algorithms, while researchers often focus more on scientific discovery rather than software engineering quality, causing testing activities to be marginal-

ized. Especially in cases where AI/ML and neuroscience software are combined, quality assessment work has not yet formed a systematic framework and standards. This not only affects the credibility of these tools in academic research but also constrains their application in clinical translation. Meanwhile, the uncertainty and data dependency of AI/ML algorithms further exacerbate the difficulty of quality verification, particularly in key dimensions such as functional correctness, transparency, and robustness.

Establishing a systematic quality assessment methodology is crucial for enhancing the credibility of AI/ML-based neuroscience software, not only concerning the reliability of academic research but also directly affecting its potential application in clinical practice. Although the field lacks a unified or formal taxonomy of software quality attributes for neuroscience applications, our review of existing studies and software use cases suggests that transparency, functional correctness, and robustness are consistently prioritized, either implicitly or explicitly, by researchers and practitioners. This observation forms the basis for the focused evaluation presented in the following chapters. Addressing these three attributes represents a critical first step toward establishing a comprehensive and reliable framework for AI/ML-based neuroscience software quality assessment

With the widespread application of AI/ML technology in the field of neuroscience, we hope that through systematic quality assessment methods, we can address the uncertainty issues brought by AI/ML algorithms in neuroscience software while ensuring transparency, making algorithm behavior and decision-making processes more traceable and verifiable, making AI/ML software systems more reliable, and ensuring that the results produced by neuroscience software are accurate, stable, and reproducible.

3

Methods

In this study, to systematically validate the critical quality attributes of AI/ML-based neuroimaging analysis software systems, we first constructed a highly controlled simulation environment. The construction of this environment follows the principle of experimental simulations, aiming to enhance internal validity by precisely controlling experimental conditions and minimizing the influence of uncontrollable external variables. In this chapter, brain functional connectivity networks are reconstructed using artificially simulated data, allowing for precise manipulation of key parameters and evaluation of software system quality attributes without interference from complex real-world variables. Specifically, a Watts-Strogatz network-based simulation data generation scheme was employed to model biologically plausible brain functional connectivity structures. By configuring parameters such as the number of nodes, average degree, rewiring probability, and salient nodes/ROIs, the complexity of the network, connection patterns, and group differences can be flexibly controlled according to research needs. This parameterized control enables flexible adjustment of data complexity and variability, providing strong support for the data generation and parameter tuning described in the subsequent simulated data section. Consequently, it allows for systematic and reproducible testing and validation of specific quality attributes, such as transparency, functional correctness, and robustness. Figure 3.1 overviews the high-level workflow of the method.

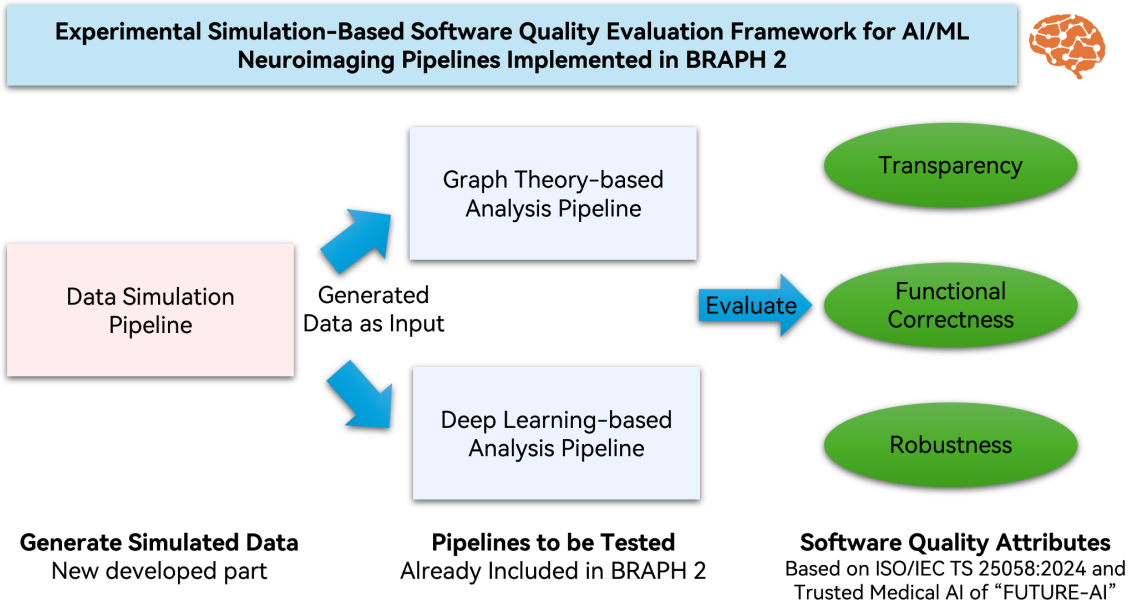

Figure 3.1: Experimental Simulation-Based Software Quality Evaluation Framework for AI/ML Neuroimaging Pipelines Implemented in BRAPH 2. Simulated brain-network data are generated via a data-simulation pipeline and then input into two analysis workflows to test the software (a graph-theory pipeline and an AI/ML-based deep-learning pipeline). The outputs from these pipelines are evaluated against three software quality attributes drawn from ISO/IEC TS 25058:2024 and a trusted medical-AI framework — namely, transparency, functional correctness, and robustness.

3.1 Software Research Method

In the ABC (Actors, Behavior, and Context) software engineering research framework [46], research strategies are classified according to two core dimensions—obtrusiveness (the degree of control researchers have over the research environment) and generalizability (the scope of applicability of research results). Among these, experimental simulations are categorized in quadrant II, representing research conducted in contrived settings. The framework uses the "greenhouse" as a metaphor for **experimental simulations**, emphasizing its essence as a highly controllable environment constructed for specific research purposes. This strategy is particularly suitable for the context of this research: first, this study constructed a dedicated environment that simulates brain data, which can control specific parameters of Watts-Strogatz networks model through the graph theory-based analysis pipeline and deep learning-based analysis pipeline; second, this environment is carefully designed to test specific software quality attributes; finally, through precise control of variables in the environment, the research can be conducted under conditions that would be difficult to achieve in real clinical environments, thereby enhancing the internal validity of the experiment and obtaining more targeted research conclusions.

3.2 Neuroimaging Analysis

The software analysis pipelines to be tested in this study focus on macroscale human brain connectome analysis. At this level, brain regions are defined as network nodes, and their connections are established through non-invasive imaging modalities. In particular, resting-state fMRI is employed to capture patterns of functional connectivity, which are subsequently represented as graph structures [47], [48]. Figure 3.2 [49] illustrates the process of constructing a brain connectome: first, a parcellation or atlas is selected to define the network nodes. Statistical dependencies between the time series of each node (e.g., Pearson correlation coefficients) are then calculated to generate a functional connectivity matrix. This matrix can be thresholded and binarized to form an adjacency matrix, or alternatively retained in a weighted and fully connected form. Finally, using the spatial coordinates of each brain region, nodes and edges can be visualized on a three-dimensional brain surface to provide an intuitive representation of the network topology. To further understand the global organizational characteristics of brain networks, the Watts–Strogatz network model is commonly employed as a comparative framework [50].

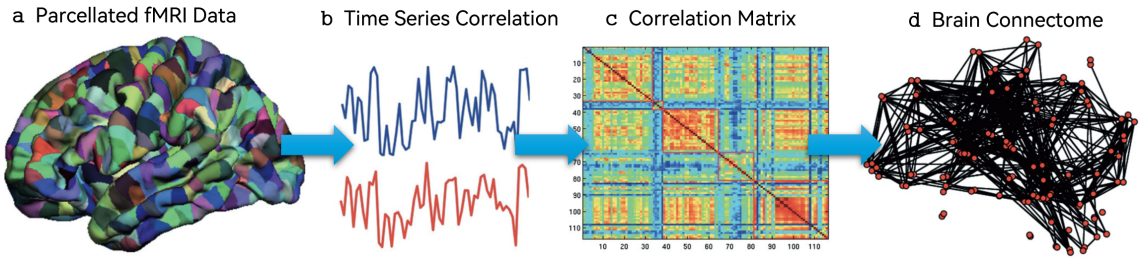

Figure 3.2: Brain connectome construction process, figure adapted from Hart et al. (2016) [49]. **a** illustrates the parcellation of resting-state fMRI data into anatomical regions, each serving as a network node. **b** describes the extraction of the representative blood oxygenation level dependent signal time series from each region and the calculation of pairwise correlations. **c** presents how these statistical associations are organized into a correlation matrix, which may be kept weighted or thresholded into a binary adjacency matrix. **d** shows the visualization of the resulting network using the spatial coordinates of regions, mapping nodes onto the brain surface and edges onto functional connections.

Figure 3.3 [51] compares regular, random, and small-world networks. Regular networks exhibit high clustering (reflecting strong local efficiency) but long characteristic path lengths (low global efficiency). In contrast, random networks show low clustering but short path lengths (high global efficiency). The Watts–Strogatz model generates networks that interpolate between these two extremes, thereby producing the hallmark small-world topology—characterized by simultaneously high clustering and short path lengths [52]. Such a small-world organization, which can be represented by Watts-Strogatz network model, has been consistently observed in

healthy human brain networks and across various neuropsychiatric conditions [53]. For example, prior studies [54], [55] indicate that both patients with Attention-Deficit/Hyperactivity Disorder (ADHD) and healthy children retain small-world properties in their functional and structural networks; however, ADHD networks tend to shift toward more lattice-like configurations, with increased local efficiency and reduced global efficiency. This topological deviation provides an important basis for applying graph-theoretical methods in the pipelines under evaluation.

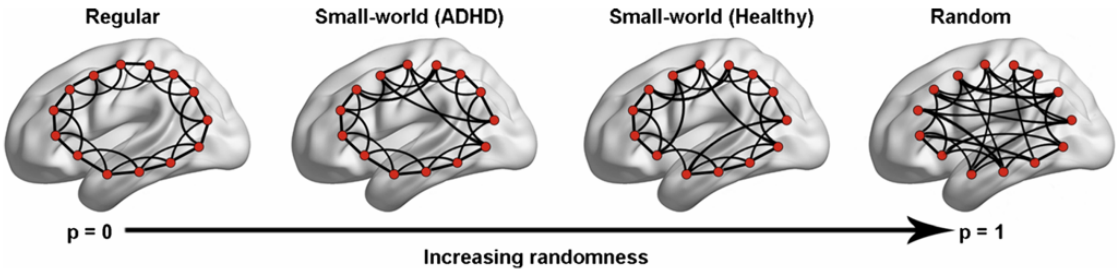

Figure 3.3: Comparison of brain network topologies with increasing rewiring probability [51]. Brain connectomes, constructed through the process illustrated in Figure 3.2, can be modeled using the Watts–Strogatz model, where the rewiring probability p interpolates between a regular lattice and a random network. Regular networks exhibit high clustering but long characteristic path lengths, while random networks show low clustering and short path lengths. For intermediate p values, the model produces networks with both high clustering and short paths—capturing the small-world topology. Such a small-world organization has been consistently observed in healthy human brain networks, whereas deviations from this pattern, such as a shift toward more lattice-like configurations, have been reported in conditions like ADHD.

In the graph theory-based pipeline, functional connectivity networks are derived from resting-state fMRI data and represented as graphs, using the Watts–Strogatz network model, where nodes represent brain regions and edges represent directed effective connectivity between them. This modeling captures key properties of brain networks, namely high clustering and short path lengths. In the deep learning-based pipeline, resting-state fMRI–derived connectivity matrices are used as input to neural network models trained in an end-to-end fashion, enabling automated feature extraction and classification of brain states.

As introduced above, two primary different types of analysis pipelines have become prevalent in contemporary brain imaging workflows: a graph theory–based workflow for identifying key brain regions, and a deep learning–based workflow for modeling and predicting connectivity patterns. However, parameter adjustments in traditional graph theory–based analysis methods are often highly dependent on researchers' prior experience and the correctness of the software implementation, making it challenging to generalize such methods across different datasets, disease types, or research contexts. While deep learning–based methods offer advantages in automatically extracting high-level features from complex brain imaging data and

often achieve superior predictive performance, they also present challenges. These include the need for large labeled datasets, limited interpretability, and high computational demands. Moreover, due to their black-box nature, ensuring the correctness, robustness, and transparency of deep learning-based models remains a significant concern, especially in critical applications such as medical diagnosis. These limitations underscore the importance of evaluating software quality systematically across both pipeline types.

As part of the research methodology, this subsection has established the methodological basis for the later application of the evaluation framework. The following subsection will introduce the background of the pipelines that will be tested throughout the study, providing the technical context for their subsequent evaluation.

3.2.1 Graph Theory-based Analysis Pipeline

Figure 3.4 illustrates the workflow of graph theory-based analysis pipeline, which can be described as follows: the entire process includes 1. preparation of fMRI data, 2. graph network construction, 3. graph theory analysis, and finally, 4. group comparison.

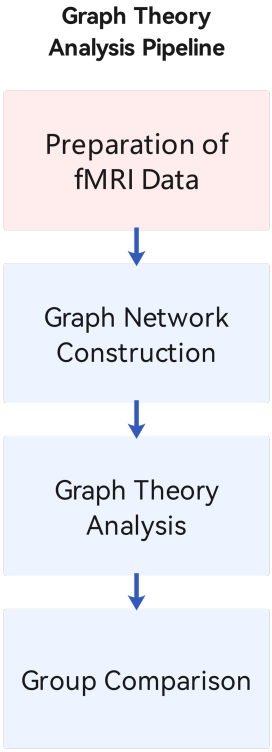

Figure 3.4: Graph Theory-based Analysis Pipeline Workflow. This workflow illustrates the main steps of graph theory–based analysis pipeline of brain functional connectivity: 1. obtain fMRI data; 2. construction of weighted undirected functional connectivity graphs from correlation matrices; 3. computation of global and local graph-theoretical metrics (e.g., clustering coefficient, path length, rewiring probability) and statistical testing; and 4. group-level comparison of network properties (e.g., Alzheimer's disease vs healthy controls), combined with brain region visualization to highlight hubs, modular structures, and connectivity alterations. This pipeline enables the quantitative characterization of brain functional network topology, facilitating the investigation of neural connectivity alterations associated with cognitive function, behavioral traits, or neurological disorders, and serves as the foundation for the graph theory-based analysis pipeline testing conducted in this study.

Using graph theory-based analysis pipeline to evaluate the topological characteristics of brain functional connectivity networks, this pipeline allows us to build weighted undirected graph networks at the individual level based on graph theory knowledge, and comprehensively analyze brain network structures through graph theory metrics (such as rewiring probability, clustering coefficient, path length, etc.), assessing the impact of disease states or behavioral variables on neural network organization.

1. Preparation of fMRI data
   First, obtain fMRI data that has undergone standardized preprocessing.

2. Graph Network Construction

The software converts these correlation matrices into weighted undirected graphs (WU), where the weight of each edge is the correlation between pairs of brain regions, outputting the functional connectivity strength between different regions. Through the software's GUI (Graphical User Interface), you can directly select the type of graph to analyze and process negative correlations (preserve/remove/take absolute value). Here, the strategy of preserving positive correlations and setting negative values to zero is adopted.

3. Graph Theory Analysis
   Calculate global and local metrics of the network using BRAPH's built-in algorithms, such as path length, connection probability, clustering coefficient, etc. At the same time, non-parametric permutation tests are used to evaluate group differences, with the option to normalize by comparison with random graphs to enhance the interpretability of metrics.

4. Group Comparison
   Perform statistical significance analysis on network metrics, including significance testing of differences between different subject groups (such as Alzheimer's disease vs healthy controls) (i.e., between-group differences). Combined with brain region visualization functionality, display high-participation nodes (hubs), modular structures, and connection changes, etc. Through graphs and statistical results generated by BRAPH, more intuitive presentation and interpretation of results can be achieved.

3.2.2 Deep Learning-based Analysis Pipeline

Figure 3.5 illustrates the workflow of the deep learning-based analysis pipeline, which can be described as follows: (1) preparation of fMRI data, (2) graph network construction, (3) data partitioning, (4) model training, and (5) feature prediction.

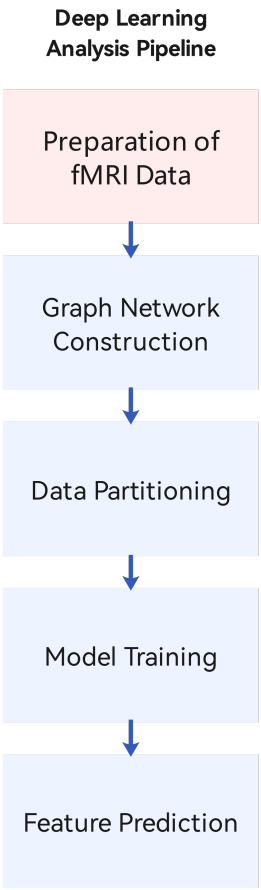

Figure 3.5: Deep Learning-based Analysis Pipeline Workflow. This workflow outlines the application of deep learning to functional-connectivity analysis, consisting of five major steps: 1. obtain fMRI data; 2. construction of subject-specific functional connectivity graphs, represented as weighted matrices; 3. data partitioning into training, validation, and test sets with fixed random seeds to ensure reproducibility and covariate balance; 4. model training using a MLP (Multilayer Perceptron) regressor, with hyperparameters, initialization, and checkpoints explicitly recorded for transparency; and 5. feature prediction, where the trained model is applied to the test set to estimate continuous measures (e.g., age, sex, cognition scores) from brain connectivity features. This pipeline enables the application of deep learning to functional connectomics for predicting individual traits, and serves as the foundation for the deep learning-based analysis pipeline testing conducted in this study.

1. fMRI Data Preprocessing
   Similar to the graph theory process, fMRI data is obtained for analysis.

2. Graph Network Construction
   Same as the graph theory process, using software to generate functional connectivity matrices for each subject. These connectivity matrices serve as input features for subsequent neural network models.

3. Data Partitioning
   A training/validation/test partitioning strategy with a fixed random seed is used to ensure reproducibility. The partitioning index and random seed are recorded and exported during partitioning, and the balance of key covariates is controlled.

4. Model Training
   A MLP regressor was used to model the connectivity features; prior to training, all hyperparameters (e.g., number of hidden layers, training epochs, optimizer/solver, batch size, learning rate) and random initialization were explicitly recorded through the interface or configuration files, and training logs and checkpoints were stored to ensure process transparency and reproducibility.

5. Feature Prediction
   After model training, the MLP regressor was applied to the test set to predict measures, such as age, sex, or cognition scores, from each subject's vectorized functional-connectivity matrix. The continuous predictions were stable and recovered the expected target patterns, with representative runs showing high consistency with ground truth.

3.3 Design of Quality Attribute Verification

In this study, the AI/ML-based software quality of BRAPH 2 is evaluated by two analysis pipelines, a graph theory–based workflow for identifying key brain regions, and a deep learning–based workflow for modeling and predicting connectivity patterns, using simulated fMRI connectivity data. The simulation datasets, generated with the Watts–Strogatz model and predefined ROIs, provide ground truth for systematic testing. Evaluation focuses on transparency, functional correctness, and robustness, assessed through GUI accessibility, recovery of expected network features, and stability under noise.

The selection of "transparency," "functional correctness," and "robustness" as quality attributes is based on their direct correspondence to the primary challenges faced by AI/ML-based neuroimaging analysis systems in real-world environments: user distrust of black-box models, potential computational errors arising from complex system function chains, and performance degradation under actual data perturbations. Consequently, to specifically validate these three software quality attributes in AI/ML-based brain imaging software systems facing multiple challenges, we designed different testing strategies for the graph theory-based analysis pipeline and the deep learning-based analysis pipeline, and experimentally verified their performance across various quality dimensions.

This research focuses on the measurement and evaluation methods of three key quality attributes. Transparency evaluates whether the software can enable non-computer specialists to understand the analytical processes and algorithmic decision logic through user-friendly interfaces, visualization outputs, and parameter recording

mechanisms. Functional correctness verifies output results (such as Watts-Strogatz property indicators) against expectations by simulating networks with known structures, ensuring reliable analytical process logic and computational accuracy. Robustness assesses whether the system can maintain output stability and accuracy when facing real-world data fluctuations by adding noise or perturbing input data.

Through this systematic design, our research not only provides a reusable quality validation mechanism for AI/ML-based neuroimaging software but also delivers quantifiable evaluation metrics for subsequent development, thereby addressing the critical shortcomings in quality control within current neuroscience software.

3.3.1 Simulated Data

In this research, in order to systematically verify and test important software quality attributes in the two analysis processes, a functional connectivity simulation data generation scheme is designed, based on Watts-Strogatz network model. And all stages of the data simulation are based on the standard connectome construction process shown in the previous Figure 3.2.

For the ground-truth topology, the Watts–Strogatz model is employed, which is shown in the previous Figure 3.3 and introduces random edge rewiring with a tunable probability to interpolate between lattice-like and random networks. The Watts–Strogatz model is particularly suitable for this purpose because it yields networks that combine high local clustering with short characteristic path lengths, closely resembling the topological organization of real human brain networks. This ensures that the simulated datasets are both biologically plausible and structurally controllable, providing an effective basis for transparent, reproducible, and rigorous evaluation of the software pipelines.

This simulation framework is based on the Watts-Strogatz model to generate network graphs, supporting customization of the following key parameters:

1. Number of Nodes
   Each node represents a brain region, and the number of nodes can be flexibly set to simulate brain parcellation schemes at different resolutions. This allows the entire simulated data to flexibly adapt to the analysis needs of different granularities.

2. Average Degree
   Each node is initially connected to its K nearest neighbors, where K is an even number, controlling the density of initial connections. This parameter determines the density of the connection matrix and can be used to simulate differences in brain states between low connectivity and high connectivity, such as comparisons between patients with neurodegenerative diseases and healthy populations.

3. Rewiring Probability

This parameter controls the degree of transition of the network from a regular structure to a random structure, defined as the probability of each edge being reconnected.

Typical settings include:

$p = 0.0$: Completely regular network (high clustering, long path)
$p = 0.1$: Typical Watts-Strogatz network
$p = 1.0$: Completely random network (low clustering, short path)

We use this feature to simulate changes in connection efficiency under different brain states. By setting different p values, we can construct two groups of functional connectivity graphs (simulating Group A and Group B), thus providing structural differences for classification tasks or between-group comparisons.

4. Salient Nodes / ROIs
In the simulated network, a set of nodes can be designated as "functionally critical regions," for example, setting 10 nodes to have higher connection strength, more cross-module connections, or stronger module centrality, to simulate "disease hubs" or "connectome fingerprints" common in real research. This feature can be used to test whether the analysis process can successfully identify key nodes or high-impact connections.

Besides, the average path length is a derived network property computed from these settings and is not directly tunable. But it is an important parameter in analyzing the Watts-Strogatz model.

- Average Path Length
Average path length is one of the important indicators measuring the efficiency of information transmission in a network, defined as the average of the shortest path lengths between any two nodes. In brain functional connectivity networks, this indicator reflects the conduction efficiency of neural signals between different brain regions, is an important component of Watts-Strogatz characteristics, and is closely related to cognitive processing speed and neural integration capability [56].

  The average path length L of a network G with N nodes is defined mathematically as:

$$L = \frac{1}{\frac{N(N-1)}{2}} \sum_{i \neq j} d(i, j) \tag{3.1}$$

  where $d(i, j)$ denotes the shortest path length between node i and node j in the network.

To better illustrate the internal logic of the simulation framework, a schematic pro-

cess flow is provided in Figure 3.6. The pipeline outlines the sequential stages of generating and managing the simulated data. Specifically, the workflow begins with (1) loading a brain atlas, where each ROI is defined as a network node. Next, (2) simulation parameters are configured, including the number of nodes, average degree, rewiring probability, and salient ROIs, based on the Watts–Strogatz model described above. Once key parameters are set, (3) functional connectivity graphs are constructed, creating Watts–Strogatz networks with biologically plausible topological features. The system further integrates a real-time visualization module, allowing users to inspect and validate the generated networks before analysis directly. Finally, (4) results are visualized and exported, with adjacency matrices and parameter metadata automatically saved in structured Excel files, ensuring transparency, reproducibility, and usability for downstream quality testing. This simulation pipeline provides a highly controllable and biomimetic environment that supports rigorous evaluation of software quality attributes such as transparency, functional correctness, and robustness, while maintaining a user-friendly and reproducible workflow.

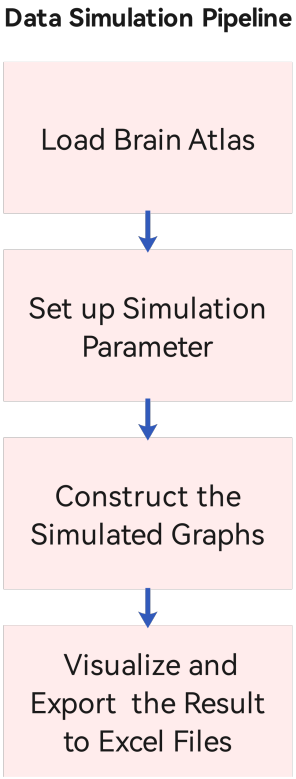

Figure 3.6: Design of Data Simulation Pipeline. This workflow outlines the generation process of biologically plausible and structurally controllable simulated functional connectivity data, based on the Watts–Strogatz network model. The pipeline consists of four main stages: 1. load a brain atlas to define the node structure of the network; 2. configure key simulation parameters including number of nodes, average degree, rewiring probability, and salient ROIs to control network topological features; 3. generate functional connectivity graphs based on the Watts–Strogatz model, with real-time visualization to support validation and quality inspection; and 4. export results, including adjacency matrices and simulation metadata, into structured Excel files for downstream use. This simulated dataset serves as a standardized and reproducible input for both the graph theory-based and deep learning-based analysis pipelines introduced in the following sections.

Through the above design, we can control the structural complexity and between-group differences of simulated connectivity data, evaluating the performance of the analysis process in terms of feature identification, model performance, and explainability. Additionally, preserving random seeds ensures that the generation process of each set of simulated data has reproducibility, facilitating stable comparisons between different experimental rounds.

3.3.2 Design of Testing Pipelines

The testing framework was designed to evaluate two neuroimaging analysis pipelines—one based on graph theory and one on deep learning—against the key software qual-

ity attributes of transparency, functional correctness, and robustness, as defined in ISO/IEC TS 25058:2024 [17] and the FUTURE-AI [26] guidelines. To ensure rigorous and reproducible evaluation, both pipelines were tested on controlled simulated datasets generated through a dedicated data simulation pipeline based on BRAPH 2.

The following table 3.1 shows the metrics that need to be tested for the two pipelines for the three attributes. In detail, for transparency, the graph-theory pipeline provided GUI-based visualization of network construction and thresholding, with all parameters and intermediate artifacts recorded and exportable, while the deep-learning pipeline offered interactive displays covering data loading, model structure, training configuration, and outcomes. Regarding functional correctness, the graph-theory pipeline was validated by checking whether predefined ROIs were consistently identified as important nodes and by verifying theoretical trends such as decreasing path length with increasing rewiring probability. In the deep-learning pipeline, functional correctness was evaluated through the agreement between predicted and ground-truth values (in this case, predicting the rewiring probabilities from the corresponding simulated network), primarily using Pearson correlation and alignment of prediction distributions with theoretical expectations. In the graph-theory pipeline, robustness was assessed by fixing algorithmic settings and repeating analyses across multiple randomly generated network instances, with the hit-rate distribution serving as the stability metric. In the deep-learning pipeline, robustness was examined by fixing the dataset while varying the model initialization of weights and biases, across independent training runs, recording the distribution of Pearson correlations to verify stability.

Testing Graph Theory–based Analysis Pipeline

The graph theory–based workflow is organized into four methodological stages—(1) fMRI data preparation, (2) graph network construction, (3) graph-theoretic analysis, and (4) group comparison, each instrumented to evaluate transparency, functional correctness, and robustness under simulation control. At a high level, the **evaluation of transparency** emphasizes user relevance at both *design-time* and *run-time*. Concretely, the transparency metric is defined by whether four aspects are satisfied: (i) accuracy of parameter control in the panel, (ii) clarity of graphical outputs for simulated data and test results, (iii) completeness of data export and analysis workflows, and (iv) thoroughness of visualized input–output information. These criteria apply across the testing of graph theory–based analysis pipeline.

1. fMRI data preparation.
   We employ the metiond data simulation pipeline based on the Watts–Strogatz model to generate connectivity matrices with controllable topology. A configuration (number of nodes, average degree, rewiring probability, and a predefined set of salient ROIs) specifies the data-generating process, different random seeds yield independent instantiations based on the mentioned configuration, while holding algorithms fixed. Here, a "subject" denotes one simulated individual; each subject is represented by a single $N * N$ functional connectivity

Table 3.1: Quality attributes and measurement examples for the two pipelines in BRAPH 2 software.

Attribute	**Goal**	**Measurement Example**	
		Graph Theory-based Pipeline	**Deep Learning-based Pipeline**
Transparency	To ensure that all analysis procedures are open, comprehensive, and understandable to users [17].	GUI shows network construction, parameters, and intermediate outputs.	GUI displays data loading, model setup, training process, and results.
Functional Correctness	To ensure that output always meets the expected or defined correctness criteria [17].	Check the correctness of simulated data and the identification of predefined ROIs.	Train MLP on simulated features, then compare predictions with ground truth.
Robustness	To ensure that the software system can stay stable and give reliable results even when facing noise [17].	Repeat analysis on random datasets; assess variability of results.	Run multiple trainings with different initializations of the model, then check stability.

Transparency is assessed through user-relevant GUI accessibility, which ensures accurate parameter control, clear graphical outputs, complete data export and workflow records, and thorough visualization of inputs and results. **Functional Correctness** is verified through controlling the same and correct simulated data. The graph theory-based pipeline accurately identifies key ROIs (hit rate $> 50\%$) and reproduces expected topological trends, while the deep learning-based pipeline achieves stable prediction–ground truth alignment ($r \approx 0.90 \pm 0.10$), confirming consistent computational reliability. **Robustness** is tested through repeated runs: graph theory evaluates the stability of ROI (hit rate $> 50\%$) across different random instantiations and keeping fixed algorithm and parameters, and deep learning examines whether the training stochasticity is controllable and performance remains stable ($r \approx 0.90 \pm 0.10$) under fixed data while model random seeds of varying initial weights and biases, and mini-batch order across multiple independent runs.

matrix generated under the specified configuration and a unique random seed. Each subject denotes one simulated individual connectivity graph; subjects are assigned to groups by design. For **transparency**, all parameters, seeds, and metadata (adjacency matrices, ROI annotations) are exposed in the GUI and exported for provenance and reproducibility.

2. Graph network construction.
   Simulated matrices are transformed into undirected graphs using user-specified rules (e.g., correlation-to-adjacency mapping, thresholding). Construction choices and their effects are visualized and logged so users can inspect data lineage and verify that grouping reflects simulation ground truth. This enforces traceable **design-time transparency** and prevents unintended preprocessing bias.

3. Graph theory analysis.
   The pipeline computes canonical global and local metrics (e.g., average shortest path length, connection density, clustering coefficient, centrality measures, modular partitions) with optional normalization against random-graph baselines. **Functional correctness** is operationalized via two checks: (i) theoretical trend conformity, metrics should follow known small-world behaviors (e.g., average path length decreases as rewiring probability increases); (ii) ROI recovery, node-importance rankings should align with predefined ROIs, quantified by a hit rate criterion defined according to (3.2).

4. Group comparison.
   Non-parametric permutation tests (with configurable iterations) assess between-group differences on selected metrics. **Robustness** is evaluated by fixing algorithms and parameters while varying only the input instantiation via different random seeds; stability is quantified by the dispersion of metric estimates and ROI hit rates across runs. Throughout, **run-time transparency** is maintained via GUI visualizations of inputs, intermediate artifacts, and outputs, together with exportable logs enabling independent verification and audit.

This methodological design isolates data initial variability from algorithmic settings, provides explicit, user-visible provenance at both design-time and run-time, and defines quantitative criteria for functional correctness (trend conformity, ROI hit rate) and robustness (cross-run stability) without relying on any specific empirical outcome.

Testing Deep Learning–based Analysis Pipeline

To ensure comparability with the graph theory–based workflow, the deep learning–based testing pipeline follows a five-stage, regression model-oriented design (the target is a continuous variable rather than a class label): (1) fMRI data preparation, (2) graph network construction, (3) data partitioning, (4) model training, and (5) feature identification. Transparency is evaluated at a high level across design-time and run-time using four user-relevant checks: (i) accuracy of parameter control in

the panel, (ii) clarity of graphical outputs for simulated data and test results, (iii) completeness of data export and workflow provenance, and (iv) thoroughness of visualized input–output information.

1. fMRI data preparation.
   We use the same data simulation pipeline to generate subject-level functional-connectivity matrices with controllable topology and a continuous ground-truth target (the rewiring probability or another synthetic scalar derived from the network). At the same time, configuration parameters (number of nodes, average degree, predefined ROIs) are explicitly set in the GUI and logged. For robustness studies, the dataset and all hyperparameters are held fixed across repeated runs; only random initial conditions of the deep learning model in training are varied. All inputs (matrices, predefined ROI) and their meta-data (parameters) are previewable and exportable to support provenance and reproducibility.

2. Graph network construction.
   For each subject, we derive a functional-connectivity representation (weighted, undirected). All construction choices (mappings, thresholds, normalization steps) are controlled via the panel, visualized for inspection, and recorded for later audit, preserving traceability from raw inputs to model-ready features. This enforces traceable **design-time transparency** by showing the situation of simulated data.

3. Data partitioning.
   The dataset is split into training/validation/test subsets using a fixed random seed to ensure reproducibility and covariate balance. Partition indices and the seed are exported. For robustness assessment, data partitions remain constant across repetitions. Note that, during model training, a separate seed governs (i) weight and bias initialization, (ii) data shuffling / mini-batch ordering, and (iii) any stochastic optimizer behavior; these choices are logged to make the sources of randomness explicit and controllable.

4. Model training.
   We train an MLP regressor on the vectorized connectivity features. Related hyperparameters (e.g., number of hidden layers and units, batch size, epochs) are specified in the GUI or config files and stored alongside training logs and checkpoints. To evaluate robustness, we repeat training multiple times while keeping the original data fixed and varying only the training random seed related to hyperparameters; thus, observed variability isolates model-internal stochasticity (initialization and mini-batch order) from data-induced effects. Training curves (loss/metric over epochs) are rendered and saved to support run-time transparency.

5. Feature identification.
   The trained model is applied to the held-out test set to produce continuous predictions. **Functional correctness** is assessed by the statistical correspon-

dence between predictions and ground truth—primarily via the Pearson correlation coefficient (r). Predicted-vs-true scatter plots and summary tables are generated to allow visual and quantitative inspection. **Robustness** is assessed by repeating the full training–evaluation cycle across seeds and analyzing the distribution of (r) (and auxiliary metrics) under identical data/hyperparameters. All artifacts (predictions, metrics, seeds, configs, logs, checkpoints) are exported to enable independent verification and audit. Throughout, **run-time transparency** can be enhanced by thorough input–output views of real-time output of results.

This testing pipeline of deep learning–based fixes the dataset, feature construction, and hyperparameters while varying only controlled sources of training stochasticity (via seeds), thereby decoupling model variance from data effects. It embeds user-relevant **transparency** at design-time and run-time through precise parameter control, clear visualizations, complete exports, and thorough input–output views. **Functional correctness** is operationalized as prediction–ground-truth agreement on held-out data (e.g., Pearson r), and **robustness** is quantified as cross-run stability of these metrics under different initializations. Complete provenance—configs, seeds, logs, checkpoints, and artifacts—supports independent reproduction and audit.

Through the above design of two pipelines, we aim to systematically identify and control potential quality risks in different pipelines through streamlined, standardized testing methods. The next section will introduce in detail the types of simulated data used and the generation methods to support the implementation of this testing system.

3.3.3 Quality Attributes

To effectively evaluate the performance of AI/ML-based systems in various neuroimaging analysis tasks in terms of quality attributes, we focus on three key quality attributes mentioned earlier: Transparency, Functional Correctness, and Robustness. These dimensions are not only emphasized in medical AI/ML-related standards such as ISO/IEC TS 25058:2024 [17] but are also considered by the FUTURE-AI [26] consensus as the foundation for trustworthy medical AI/ML software systems.

Transparency: Explanation and Visualization of Analysis Process

In clinical scenarios, when doctors and researchers without computer backgrounds use AI/ML-based software systems to perform brain network analysis in patients with brain disorders, if the software cannot provide visualization information and parameter explanations for key analytical steps, when the software generates graph theory metrics (including number of nodes, average degree, rewiring probability, and ROIs, etc.), they cannot track each step of the calculation process or explain its meaning. This may create difficulties for doctors and researchers without computer backgrounds in understanding the decision-making process behind the algorithm's

conclusions, potentially reducing trust in the software's output.

BRAPH 2 provides a GUI framework and complete analysis process recording mechanism can be developed basing on the code interface, making the analysis process highly traceable even for non-programming users. During software testing, parameters in the analysis pipeline can be manually changed through the graphical interface, automatically generating files that record every step's settings and calculation parameters, while also supporting the export of interactive result graphs, making the settings, calculation logic, and result display of each analysis step visible, checkable, and reproducible. Additionally, the function of preserving random seed settings also enhances result reproducibility.

From a software engineering perspective, visualization is not merely a matter of interface presentation but a means of achieving transparency [57]. By incorporating different types of metadata into the GUI, transparency can be systematically represented: data source descriptions enhance accuracy, image annotations improve clarity, process records ensure completeness, and result visualization reflects thoroughness [58]. This design enables users to trace the input, processing, and output of data at every analytical step, allowing for a quantitative assessment of the software's perceived transparency and fostering better understanding and trust.

It is not the case that all metadata output and visualization represent transparency; the evaluation of transparency should consider user relevance at both the design-time and run-time stages [58]. Regardless of whether it involves parameter configuration, data analysis, or result presentation, the system should, through a user-friendly graphical interface, enable users without a computer science background to easily inspect the inputs, outputs and results of each step.

Specifically, the transparency metric can be defined by examining whether the following four aspects are achieved: the accuracy of parameter control in the panel, the clarity of graphical outputs for simulated data and testing results, the completeness of data export and analysis workflows, and the thoroughness of visualized input and output information.

At both design-time and run-time, transparency is evaluated via four user-relevant checks applied across the data simulation pipeline and the two tested pipelines (graph theory and deep learning):

- Accuracy of parameter control — The GUI exposes and records all controllable parameters with exact seeds and diffs: Watts–Strogatz settings (number of nodes, average degree, rewiring probability, ROI list) for simulation; thresholding rules and graph-construction options for the graph-theory pipeline; and MLP hyperparameters (e.g., batch size, epochs) and fixed data partitions for the deep-learning pipeline.

- Clarity of graphical outputs — Watts-Strogatz network model is visually apparent in $5 * 5$ grids (data simulation); the graph-theory pipeline highlights user-defined salient ROIs and renders metric trends (e.g., path length vs.

rewiring probability); the deep-learning pipeline shows training curves and predicted-vs-true scatter plots, making performance and behavior immediately interpretable.

- Completeness of data export and workflow records — Full provenance is exported for audit: simulation configs (parameters, seeds), adjacency matrices, ROI definitions; graph-theory metric tables, threshold/normalization settings, permutation-test configs and ROI rankings; deep-learning partitions (indices, seed), configs, training logs, checkpoints, and test predictions.

- Thoroughness of visualized input–output information — Inputs and outputs are co-visualized with links to artifacts: input graphs/features and corresponding outputs (e.g., ROI hit rate, path-length summaries) in the graph-theory pipeline; and vectorized connectivity inputs with evaluation metrics (e.g., Pearson r) in the deep-learning pipeline, enabling end-to-end traceability without developer intervention.

Meeting these dimensions collectively ensures the software's interpretability, traceability, and user trust, achieving a truly meaningful assessment of transparency.

Functional Correctness: Result Consistency and Logical Reliability

Functional correctness is a core attribute ensuring that analysis results are scientifically reliable and reproducible. For the same input data, if an AI/ML-based software system outputs significantly different brain network metrics (such as path length, connection probability, clustering coefficient, etc.) across different versions or operating systems, or if changes in random seeds in the same execution environment occur, these may lead to inconsistent research conclusions.

The graph theory-based analysis process is based on clearly defined topological metrics in mathematics (including number of nodes, average degree, rewiring probability, and ROIs) and standard graph structure transformation methods (including correlation matrix construction and threshold processing), making its core computational process theoretically verifiable. When testing on simulated data, one can verify whether network properties are accurately calculated using the Watts-Strogatz network model with known structures and can also check the reasonableness of outputs under boundary condition inputs (all-zero matrix, all-one matrix, single-connection graph) or specific scenarios. For the graph theory-based analysis pipeline, functional correctness was quantitatively assessed using the *hit rate* of correctly identified predefined brain regions, representing the level of agreement between the computed node importance ranking and the known salient ROIs:

$$\text{Hit Rate} = \frac{N_{\text{correct}}}{N_{\text{total}}}, \tag{3.2}$$

where N_{correct} denotes the number of correctly identified predefined ROIs, and N_{total} is the total number of ground-truth ROIs. A high hit rate (typically more than 50%) indicates that the algorithm reliably detected the majority of salient regions across

repeated simulations, confirming the functional correctness and internal consistency of its graph-theoretical computations.

In the deep learning-based analysis pipeline, correctness was evaluated by training a multilayer perceptron regressor on the simulated connectivity data and comparing the model's predictions with the known target values. A high Pearson correlation between predicted and ground truth values in a controlled running could be confirmed that the model accurately captured the intended input–output mapping. For the deep learning-based analysis pipeline, functional correctness was evaluated through the statistical correspondence between predicted and ground-truth outputs using the Pearson correlation coefficient ($r \approx 0.90 \pm 0.10$). The narrow variation range of r across repeated runs demonstrates that the regression model maintained stable predictive accuracy and produced reproducible mappings between simulated input features and target values.

Together, these two indicators confirm that both the graph theory-based and deep learning-based pipelines produced functional correctness and reproducible outputs under controlled experimental conditions.

Robustness: Resistance to Data Anomalies and Uncertainty

In clinical scenarios, patient fMRI data often experiences mild distortion due to head movement, signal drift, and other factors. Unlike functional correctness, in this case, the input data has changed. If the graph construction algorithm is overly sensitive to data noise, the final network structure might be completely different, potentially misleading disease prediction. Existing research has found that the same software (such as FreeSurfer) shows significantly different failure rates when processing brain regions under different scanning environments, or that simply changing the operating system might lead to significant changes in neuroimaging analysis results. Unlike functional correctness—where inputs are held fixed—robustness explicitly concerns how outputs change when inputs or the learning process are perturbed. Concretely, the criterion focuses on how much a pipeline's key metric varies under data variability and algorithmic randomness, while all other settings are held fixed.

For the graph-theory analysis pipeline, it is related to data variability or input perturbations. Under the same data simulation pipeline, we instantiate multiple simulated brain networks with different random seeds so that graphs share identical constraints but differ in connectivity realisations. The graph-theory pipeline is executed on each instance and the hit rate of predefined ROIs is recorded per run. Distributional stability of the hit rate across runs reflects tolerance to structural changes in the input. We evaluate the distribution of ROI hit rates across random graph instantiations; robustness is satisfied when dispersion is small and the *lower* confidence bound remains above a predefined correctness baseline (hit rate $> 50\%$).

For the deep-learning analysis pipeline, it is related to algorithmic randomness or process perturbations. With the dataset and hyperparameters fixed, we repeatedly train the same neural model while varying the random seed that initialises all train-

able parameters (initial weights and biases); in practice, this seed also governs data shuffling / mini-batch ordering and, when applicable, optimiser trajectories. For each training, we record the Pearson correlation between predictions and ground truth. Stability of the correlation across runs reflects resilience to stochasticity inherent to learning. we evaluate the distribution of Pearson correlations across independent trainings with different seeds (affecting initial weights and biases, and shuffling); robustness is satisfied when dispersion is small and performance remains stably high ($r \approx 0.90 \pm 0.10$) under fixed data.

In both cases, a robust system should maintain consistently high performance across repeated trials, with predictable changes under different initial variability, rather than abrupt drops or unstable fluctuations. Such behavior reflects the system's ability to adapt to different data or process model uncertainty, a condition frequently encountered in real-world neuroimaging scenarios.

4

Results

This chapter presents the experimental validation of the proposed quality evaluation framework for AI/ML-based neuroimaging software, addressing both the graph theory-based and deep learning-based analysis pipelines implemented in the BRAPH 2 framework. The code of evaluation framework can be acquired online [59], it is able to evaluate the software system's transparency, functional correctness, and robustness through controlled simulations, highlighting how the combined methodologies complement each other in assessing quality. The results section first demonstrates the design, usability, and transparency of the custom GUI, followed by the structure and adaptability of the data simulation pipeline. It then details the validation outcomes for the graph theory-based analysis, including its ability to reliably identify predefined key brain regions, and the deep learning-based analysis, which extends the evaluation by testing the system's capacity to handle the randomness of the algorithm itself and complex predictive models. Together, these results provide a comprehensive view of the framework's performance across distinct AI/ML approaches, directly reflecting the thesis objective of establishing a systematic, simulation-based methodology for assessing the quality of neuroimaging software.

4.1 Control Panel and Basic Functions Validation

This section demonstrates the validation of the control panel design and the verification process of the fundamental functions of the system.

To enhance the usability and transparency of the software, we designed and implemented a GUI, as shown in Figure 4.1.a, based on the BRAPH 2 framework. All GUIs are user-relevant and easy to access important information about inputs and outputs. This interface facilitates the generation of simulation data and supports subsequent analysis based on graph theory-based and deep learning-based pipelines.

To validate the transparency of the system's graph network analysis capabilities, we implemented data visualization functionality. The software generated and utilized simulated data based on the Watts–Strogatz network model, covering various rewiring probability p settings. Through controlled variation of the rewiring probability, which increased evenly from 0 to 1, precise network topologies were generated, as illustrated in Figure 4.1.b.

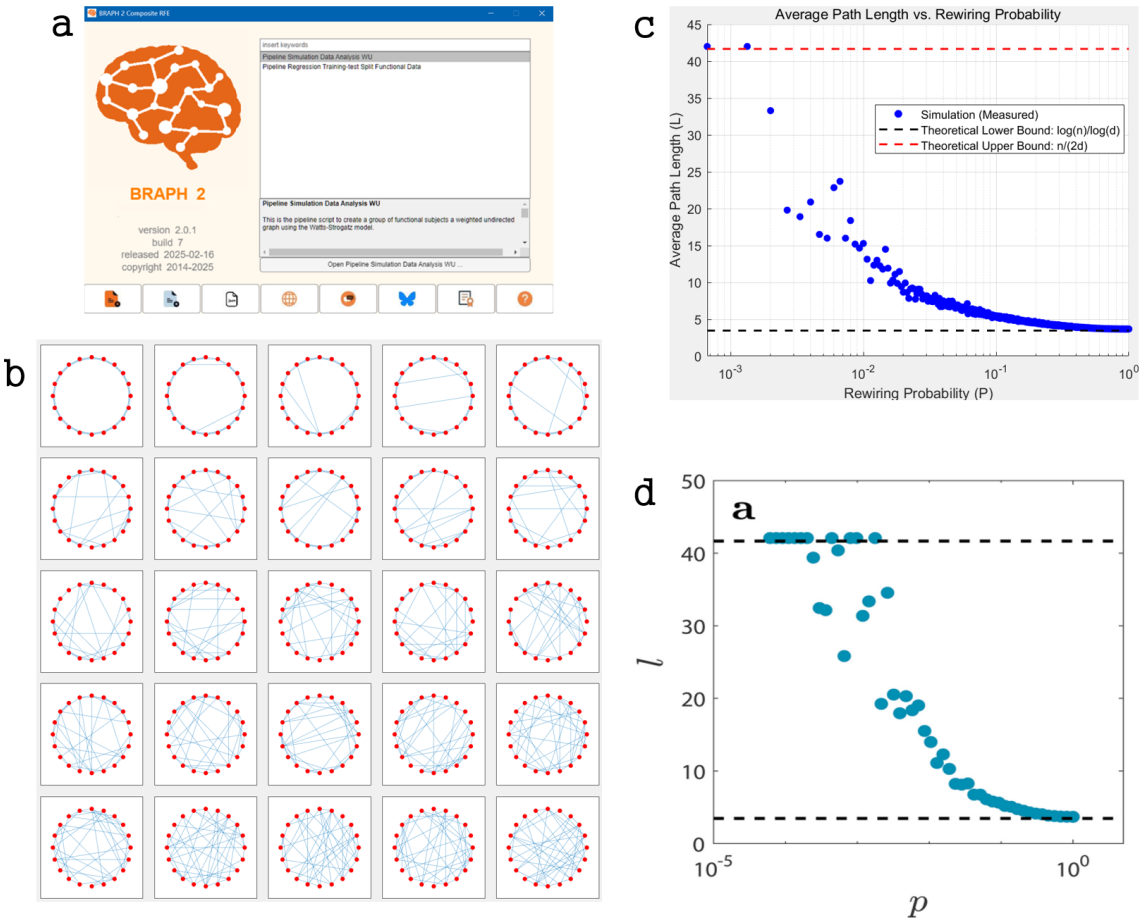

Figure 4.1: Control panel and basic functions. **a** shows a custom-designed GUI built using BRAPH 2 for generating simulation data, supporting brain network simulation, and configuration of analysis pipelines (graph theory-based or deep learning-based). **b** is an example of Watts–Strogatz network topologies generated by the data simulation pipeline in **a**, utilizing the Watts-Strogatz model with different rewiring probabilities, produced directly from the GUI to validate its functionality. A core topological feature of this model is the effect of rewiring probability on the network's average path length, which is quantitatively examined in panel **c**. **c** provides an interactive plot generated using BRAPH 2's built-in graph-theoretic calculation function for average path length, illustrating how this metric varies with rewiring probability. In connection with the small-world network topologies shown in panel **d** demonstrates the theoretically expected decrease in average path length as rewiring probability increases in the Watts–Strogatz model [60], confirming that the system can correctly compute and reflect this core topological feature.

To verify the functional correctness of the system's graph network analysis capabilities, we developed system tools that calculate the average path length under different p values, presenting the relationship between rewiring probability and average path length through interactive graphics. The results, as shown in Figure 4.1.c, demonstrate that as p increases, the average path length gradually decreases, which is consistent with the theoretical model behavior depicted in Figure 4.1.d [60].

For fundamental functions of the testing methodology, Figure 4.1 illustrates the GUI and basic functionalities we implemented.

4.2 Data Simulation Pipeline

This section introduces the data simulation pipeline we developed and utilized for testing, which is fundamentally based on the Watts–Strogatz model. To enhance user interaction experience and ensure reproducibility of results, we designed and implemented a dedicated GUI through which users can intuitively configure key parameters for network generation, including the number of nodes, average degree, and rewiring probability.

Through this GUI, we implemented flexible parameter configuration, enabling users to precisely control the structural complexity of simulated data, thereby accommodating network structure requirements under various experimental conditions, which ensures accurate parameter control, clear graphical outputs, complete data export and workflow records, and thorough visualization of inputs and results. Specifically, users can customize the number of nodes, rewiring probability, average degree, salient nodes/ROIs, and number of subjects, with the concepts of these parameters explained in detail in Chapter 3.3.2. Additionally, we integrated a real-time visualization module that automatically renders the generated network topology, assisting users in rapidly assessing network validity and structural characteristics prior to analysis.

To achieve comprehensive process tracking and data reusability, we also implemented an automatic data export function that saves the simulated adjacency matrices and all associated parameters in a structured Excel file format.

Through this simulation data pipeline, we successfully established a highly controllable and biomimetic data generation environment. For the data simulation pipeline, Figure 4.2 illustrates the whole workflow.

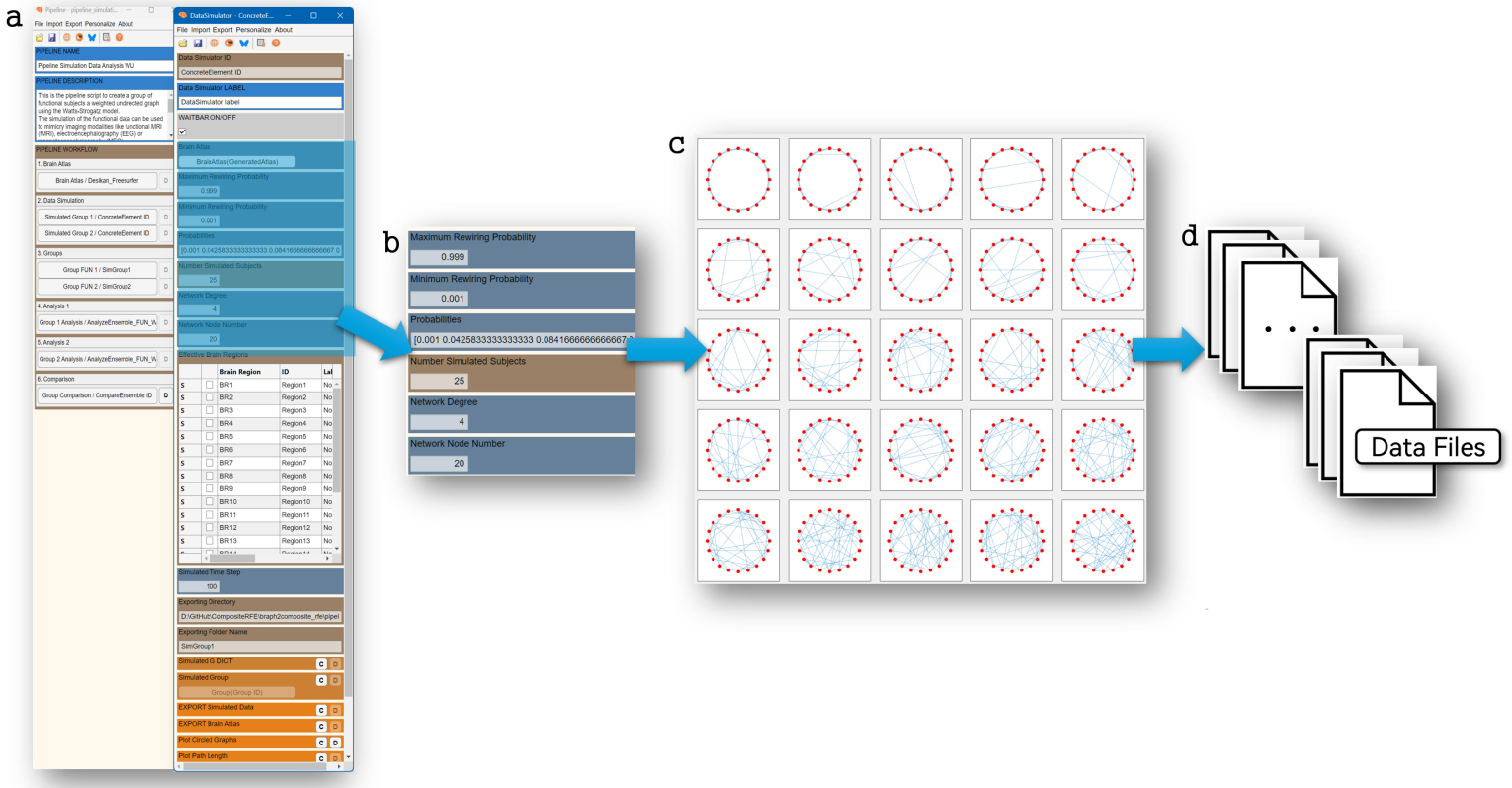

Figure 4.2: Data Simulation Pipeline. The custom-built data simulation pipeline is designed for generating functional brain connectivity data with configurable structural properties. Through a user-friendly GUI, users can flexibly set parameters such as the number of nodes, average degree, rewiring probability, number of subjects, and salient ROIs to generate functional connectivity matrices with Watts–Strogatz model properties. The system supports real-time network visualization and exports all data and settings as Excel files, ensuring transparency, reproducibility, and suitability for downstream quality testing.

4.3 Graph Theory-based Analysis Pipeline

This section demonstrates how the simulated datasets constructed using the Watts–Strogatz network model were utilized to test the software attributes of the graph theory-based analysis pipeline within the software system. Following the parameterized data simulation framework proposed above, as shown in Figures 4.3.a and 4.3.b, we constructed two functional connectivity network groups with distinct structural characteristics. Each network group had 25 subjects and comprised 20 nodes with an average degree of $D = 4$, but with different rewiring probabilities: Group 1 was configured with $p = 0.2$ to simulate a more regular Watts–Strogatz structure, while Group 2 was configured with $p = 0.8$ to reflect connection patterns more closely resembling random graphs.

During the data generation process, we manually designated 8 specific nodes (red dots) as "controlled brain regions" or ROIs to test whether the analysis pipeline possesses the capability to identify significant structural differences in the beginning. Subsequently, we employed the built-in graph theory-based analysis pipeline in the BRAPH 2 software to conduct computational and visual analyses on both groups, focusing on key topological metrics including average path length, average degree, modular partitioning, and node centrality. The analysis results indicate that the system can accurately identify the high-participation nodes I predetermined and correctly reflect the structural differences between the two network groups.

As illustrated in Figure 4.3.c, this example result, randomly selected from multiple simulation runs, demonstrates the outcome of testing the functional correctness of the graph theory-based analysis pipeline. In this run, the software accurately identified all eight predefined important brain regions among the top eleven ranked nodes based on network metrics. This confirms that, under controlled simulation settings, the pipeline can correctly recover the ground-truth regions specified during data generation.

Figure 4.3.d presents the results of a robustness evaluation in which the algorithm was kept fixed, but the simulated datasets varied randomly in each of 100 runs. The plot shows, for each run, the number of predefined ROIs correctly identified among the top eight ranked nodes. Across these trials, a high hit rate (94.25%) was maintained despite the variability of input data, indicating that the pipeline's performance remains stable when facing differences in network instances generated under the same parameter constraints, which presents the results of a robustness evaluation of AI/ML-based software.

For the testing results of the graph theory-based analysis pipeline, Figure 4.3 demonstrates that the software was able to accurately detect the functionally important brain regions specified by the user in the vast majority of cases. This validates the effectiveness of the simulation-based testing scheme proposed in this study for evaluating the software quality attributes. Moreover, the stable and reproducible performance of BRAPH 2 across multiple iterations highlights the reliability of its graph theory-based analysis module.

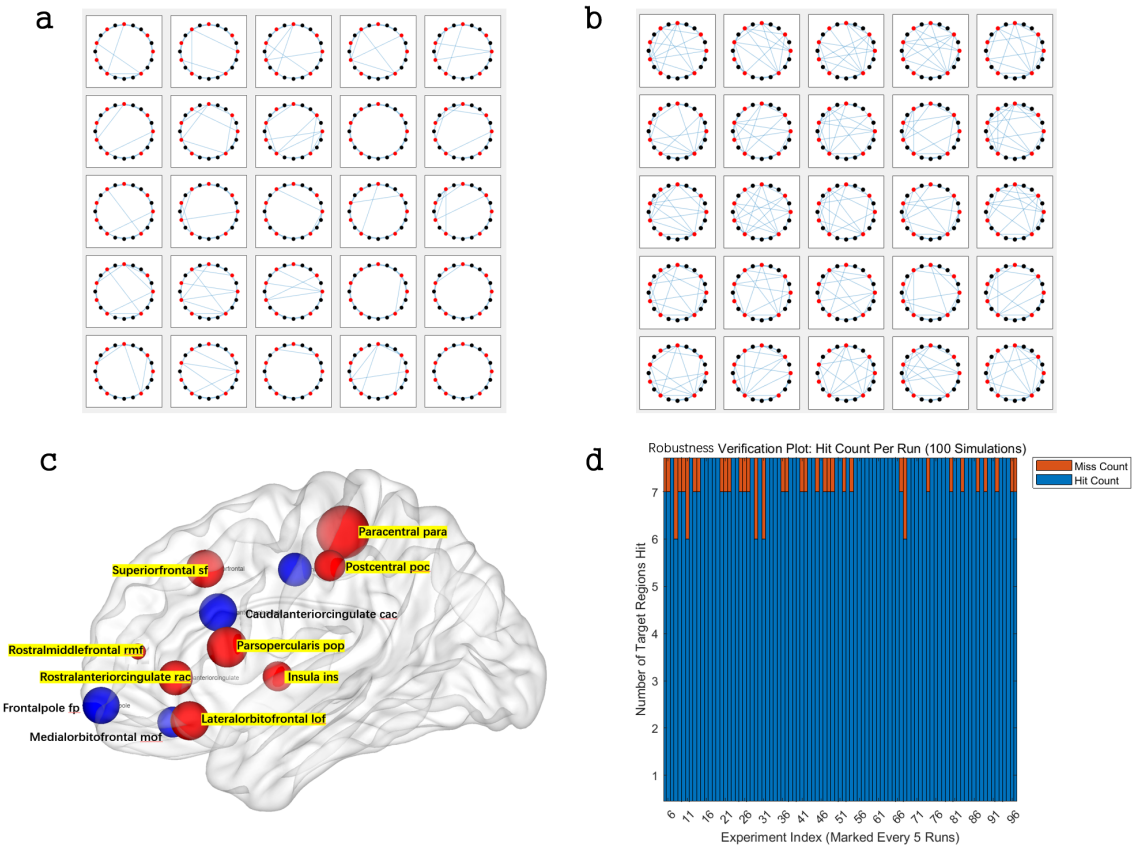

Figure 4.3: Graph Theory-based Analysis Test Using Simulated Data. a and **b** display two groups of simulated functional brain networks generated using the Watts–Strogatz model, with different rewiring probabilities ($p = 0.2$ for Group 1 and $p = 0.8$ for Group 2), reflecting structural differences between more regular and more random network topologies. Each network contains 20 nodes and highlights 8 predefined salient ROIs (red nodes) used to evaluate the analysis pipeline's ability to recover the ground truth. **c** shows the outcome of the functional correctness test using simulated data. The figure presents a representative result from a single simulation run in which the software successfully identified all eight predefined important brain regions (highlighted in yellow) among the top eleven ranked nodes based on graph-theoretic metrics, illustrating the pipeline's ability to correctly recover the ground-truth ROIs specified during data generation. **d** shows the robustness evaluation results obtained from 100 independent simulation runs, where the analysis algorithm was fixed but the input datasets were randomly generated for each run. The bar plot reports the number of predefined ROIs correctly identified among the top eight ranked nodes in each run. The consistently high hit rate (94.25%) across varied inputs demonstrates the pipeline's stable performance under controlled yet data-variable conditions.

4.4 Deep Learning-based Analysis Pipeline

This section demonstrates the transparency, functional reliability and robustness of the software system under dynamic data conditions. The deep learning-based pipeline was implemented and tested using simulated brain connectivity data. This testing pipeline was designed not only to perform predictive modeling but also to demonstrate system transparency and robust performance through repeated trials.

Figure 4.4.a illustrates the GUI of the implemented deep learning-based pipeline, designed to make the modeling process transparent and user-friendly by clear graphical outputs and thorough visualization of inputs and results. Users are able to load simulated datasets, configure hyperparameters such as the number of hidden layers, training epochs, solver type, and batch size, and inspect model configurations before training.

To ensure the pipeline's interpretability from input to output, Figure 4.4.b visualizes a subset of 25 simulated brain graphs (out of 500 total), selected across the full range of rewiring probabilities ($p = 0.05$ *to* 0.95). Each graph corresponds to a synthetic subject and follows a Watts–Strogatz model. The visualization shows the effective ROIs in red, which were embedded into the network with higher connectivity. This visual validation allows users to confirm that the simulated data distribution aligns with expectations and serves as further evidence of the pipeline's design-time transparency.

As illustrated in Figure 4.4.c, this example result, taken from one representative run of the deep learning-based pipeline, demonstrates the outcome of testing the functional correctness attribute. In this run, a multilayer perceptron regressor was trained on simulated brain connectivity data, and the predicted values closely matched the ground truth values. The scatter plot shows points tightly clustered along the identity line ($y = x$) with a Pearson correlation coefficient of 0.90. This alignment confirms that, under the given controlled conditions, the model was able to learn and reproduce the predefined mapping from input networks to target values, thereby satisfying the functional correctness requirement in a single execution.

Figure 4.4.d presents the robustness evaluation of the same pipeline, where the algorithm was kept unchanged but the training process was repeated 100 times with different random initializations of weights, biases, and mini-batch order to the neural network training procedure. Across these independent runs, the Pearson correlation remained consistently high (average $r = 0.8380$) with low variance, indicating that the model's predictive performance is stable despite the inherent stochasticity of the learning algorithm.

For the testing results of the deep learning-based analysis pipeline, these results validate the functional correctness, robustness, and capability of delivering stable and interpretable outputs even in the presence of internal variability and random optimization, which are common in real-world AI/ML-based neuroimaging systems.

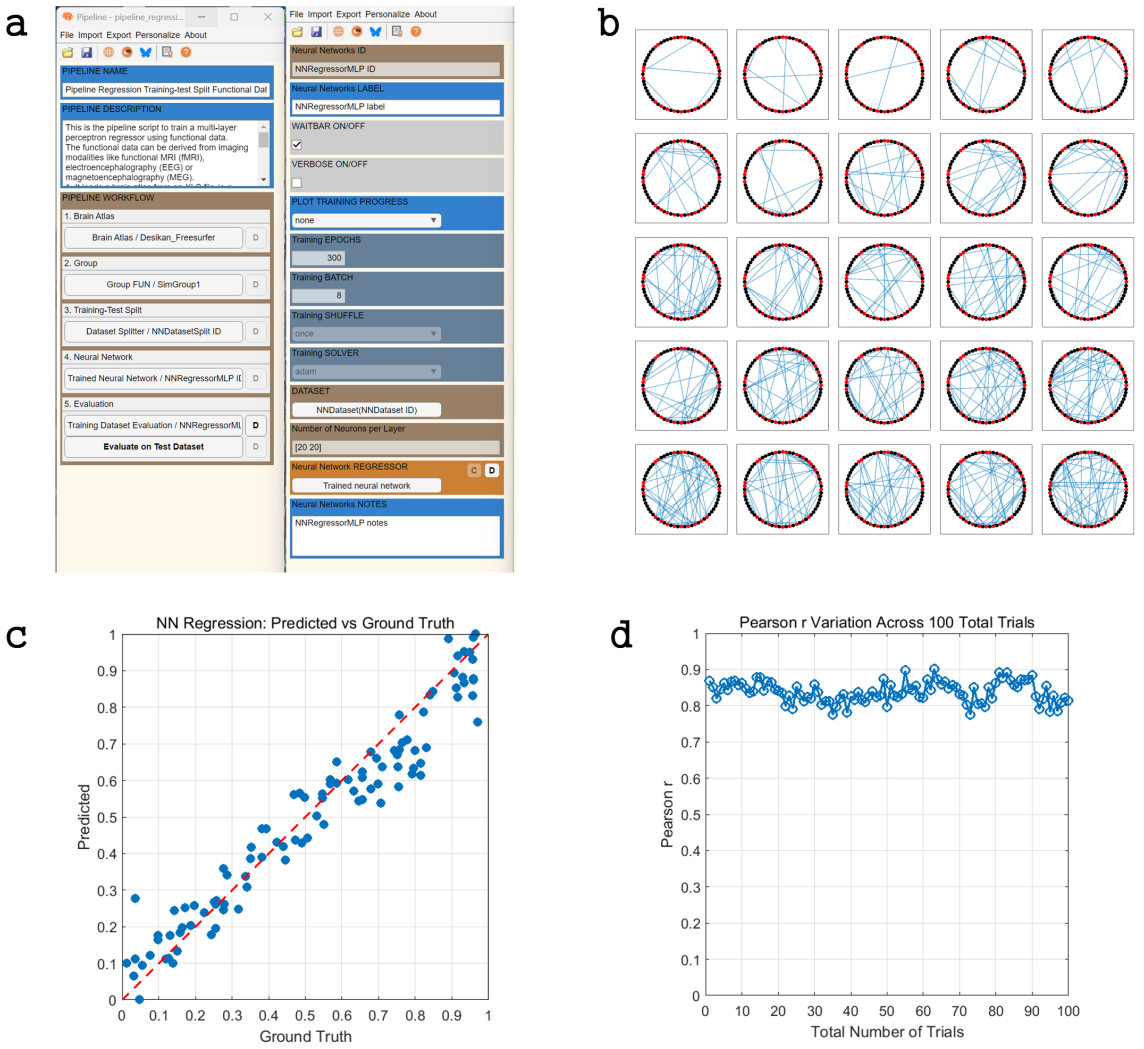

Figure 4.4: Deep Learning-based Analysis Test Using Simulated Data. **a** illustrates the user interface of the deep learning-based pipeline allows full control over data loading, model configuration, and training parameters, supporting transparency and reproducibility in the modeling process. **b** depicts a subset of 25 simulated brain graphs that illustrate the variation in connectivity across different rewiring probabilities p, with red nodes marking the predefined effective ROIs, offering visual verification and transparency of the data distribution. **c** presents the results of the functional correctness test for the deep learning-based pipeline, where the scatter plot compares predicted and ground truth values of rewiring probabilities from a representative training run of a multilayer perceptron regressor. Data points align closely with the identity line ($y = x$), and the Pearson correlation is 0.90, confirming that the model accurately reproduced the target outputs for the given inputs in a single controlled execution. **d** demonstrates the robustness evaluation of the deep learning-based pipeline based on the algorithm's inherent stochasticity. The plot shows the Pearson correlation coefficients from 100 independent runs using the same dataset but different random initializations and stochastic training sequences. The consistently high correlations (average $r = 0.8380$) and low variance demonstrate stable predictive performance despite variability introduced by the learning process itself.

5

Discussion

This chapter provides a systematic discussion centered on the core research question—how to effectively evaluate the quality attributes of AI/ML-based neuroimaging software to ensure reliable performance in real-world neuroimaging analysis pipelines. Building on ISO/IEC TS 25058:2024 [17] and the six principles of FUTURE-AI [26], an evaluation framework is introduced and justified around three pillars: transparency, functional correctness, and robustness, supported by reproducible metrics and empirical demonstrations in a controlled simulation environment. The methodological significance and usability enhancement pathways are examined from both the developer and end-user perspectives, with additional consideration of the framework's generalizability across multimodal data, cross-platform tools, and multi-task settings. Potential threats to validity are then systematically identified, with mitigation strategies proposed across the four dimensions of construct validity, internal validity, external validity, and conclusion validity, thereby strengthening the credibility and generalizability of the findings. Finally, the boundaries and norms for the use of generative AI in this study are clarified, ensuring full traceability and compliance throughout the process and providing a methodological foundation for future validation with multi-center real-world data and clinical applications.

5.1 Software Quality Evaluation

5.1.1 Framework and Objectives

Guided by the central research question: **How can quality attributes of AI/ML-based neuroscience software be effectively evaluated to ensure reliable performance in real-world neuroimaging analysis pipelines?** This section outlines the rationale and objectives for constructing the quality evaluation framework.

High-quality software is a prerequisite for ensuring the credibility and reproducibility of neuroimaging research. As AI/ML methods become increasingly integrated into brain imaging analysis, factors such as algorithmic uncertainty, data variability, and the high-stakes nature of medical decision-making make quality evaluation not merely a tool for technical optimization but an essential safeguard for the robustness of research conclusions and the clinical reliability of results.

To address the research question systematically, this study draws on the ISO/IEC TS 25058:2024 technical specification and the six principles of FUTURE-AI [26], identifying three core quality attributes—Transparency, Functional Correctness, and Robustness—as the pillars of the evaluation framework. These attributes were selected for their strong relevance to AI/ML-based neuroimaging software and their direct impact on trustworthiness in both research and clinical contexts.

The aim of this framework is to provide a reproducible, quantifiable, and interpretable pathway for systematically verifying AI/ML-based neuroimaging analysis software. By using simulated data to control variables, the framework isolates each of the three attributes and defines appropriate metrics for them. This approach enables precise assessment, offers clear answers to the research question, and establishes quality benchmarks to guide future real-world applications.

5.1.2 Transparency

According to ISO/IEC TS 25058:2024 [17], transparency is defined as the ability of an AI system to present its processes and results in an open, comprehensive, and user-understandable manner, ensuring that users can trace and review the system's decision logic. In the context of this study, this definition is refined to mean that, within AI/ML-based neuroscience software, all stages of model analysis must be accessible, comprehensible, and operable—even for end-users without programming expertise, such as neuroimaging researchers.

In medical and neuroimaging AI/ML-based software, transparency extends beyond making source code available or providing complete technical documentation. It requires that every stage of the workflow—data processing, model inference, feature extraction, and result generation—offers clear visual feedback and parameter traceability. ISO/IEC TS 25058:2024 [17] treats transparency as a foundation for traceability and auditability, while the FUTURE-AI [26] principles of Traceability and Explainability stress that systems must provide interpretable decision pathways. This ensures users not only know what the outcome is, but also understand why the system produced that outcome. In high-risk medical scenarios, such as when clinicians evaluate lesion significance or functional areas, this level of transparency is a decisive factor in whether AI recommendations are trusted and adopted.

Design-time Transparency

During software testing methodology design, transparency was ensured through a custom-built GUI that integrates a control panel with explicit parameter settings. As shown in Figures 4.1.a, 4.2, and 4.4.a, the interface enables users to directly configure key parameters for network generation, analysis, and modeling (e.g., number of nodes, average degree, rewiring probability, hidden layers, training epochs, optimizer type, batch size). By providing complete visibility into model configurations and data structures, the GUI reduces the "black box" nature often associated with AI/ML-based systems and allows users to validate settings, replicate experiments, and debug workflows before execution. As further described in Chapter 4,

it was designed to enhance usability and transparency, enabling even users without programming expertise, and thereby serving as clear evidence of design-time transparency.

Transparency was also ensured through explicit parameter recording and data flow, which are based on traceable and understandable procedures. As illustrated in Figures 4.1.b, 4.2, 4.3.a, 4.3.b, and 4.4.b, the pipeline proceeds from Watts–Strogatz model parameter configuration to network topology generation and finally to Excel-based metadata export. The exported file includes all relevant information, such as network generation parameters, random seeds, adjacency matrices, and annotated valid ROIs, thereby allowing analyses to be reproduced and audited in full compliance with the ISO/IEC TS 25058:2024 [17] definition and FUTURE-AI [26] requirement for transparency.

Run-time Transparency

During execution, transparency is maintained through interactive visualisation and interpretable result outputs. In the functional correctness tests, which can be illustrated in Figure 4.3.c, predefined ROIs are highlighted in yellow, enabling users to visually assess how well the identified brain regions correspond to the ground truth.

In addition, Figures 4.3.d, 4.4.c, and 4.4.d present outputs that combine quantitative metrics—such as hit rate and Pearson correlation coefficient—with graphical representations of prediction-to-ground-truth alignment and performance distributions across multiple runs.

These visualisations provide intuitive decision support, ensuring that the outputs are not only usable but also understandable and verifiable.

Measurement Example

In practice, transparency was quantified with four user-relevant indicators spanning design-time and run-time. First, parameter-control accuracy: the control panel exposes and validates all configurable options of the Watts–Strogatz model (e.g., number of nodes, average degree, rewiring probability, ROI list), graph-construction/thresholding rules, and MLP hyperparameters; it records random seeds and enables side-by-side comparisons of configurations before and after generation, so users can precisely verify what was run. Second, clarity of graphical outputs: the Watts–Strogatz model is made visually transparent in $(5 * 5)$ grids—(i) in the graph-theory pipeline by contrasting groups with low vs. high rewiring probability, user-defined salient ROIs are highlighted with red circles for immediate interpretability, and (ii) in the deep-learning pipeline by ordering graphs with increasing rewiring probability; all figures (predicted-vs-true scatter, multi-run distributions) include unambiguous legends, axes, and units. Third, completeness of data export and workflow provenance: the system automatically exports structured artifacts (simulation configs and seeds, adjacency matrices, ROI definitions; graph-theory metric tables, threshold/normalization settings, permutation-test configs, ROI rankings; deep-learning partitions and seeds, configs, training logs, checkpoints, and test predictions) to support full

reproduction and audit. Fourth, thoroughness of visualized input–output information: dashboards co-visualize inputs (simulated graphs/features with metadata) and outputs (ROI hit rate, path-length summaries, Pearson r) with direct links to underlying files, allowing users—without developer assistance—to inspect intermediate and final artifacts and verify the system's behavior end-to-end.

In this implementation, every step of the analysis process—both intermediate and final outputs—can be accessed and verified through the user-friendly GUI. Everything displayed in the GUI is relevant to the user's requirements [57]. This design aligns with the ISO/IEC TS 25058:2024 [17] requirement for transparency measurement: users must be able to directly access and verify system behaviour without relying on developer intervention.

5.1.3 Functional Correctness

According to ISO/IEC TS 25058:2024 [17], functional correctness refers to a system's ability to consistently produce outputs that meet predefined or expected correctness standards, and to maintain this standard reliably across repeated executions. As a core sub-attribute of Functional Suitability, functional correctness not only requires logical accuracy of results but also demands stability and verifiability under varying conditions.

In neuroimaging analysis, this means that when the analysis target and conditions are known, and the ground truth is clearly defined, the system must be able to accurately identify the predefined key brain regions or correctly complete the specified analytical tasks. This capability directly affects the reliability of research findings and the trustworthiness of clinical decisions. The Usability and Fairness principles in the FUTURE-AI [26] framework further emphasise that system outputs should remain consistent and dependable across different datasets, populations, and task conditions.

Basic Function Validation

During the early development stage, basic functionality tests were used to verify whether the software's core algorithms could produce results consistent with theoretical expectations under controlled simulation conditions. Figures 4.1.c and 4.1.d show that in networks generated using the Watts–Strogatz model, the average path length decreased as the rewiring probability p increased, fully matching theoretical predictions. The calculated results preserved the expected monotonic trends in topological metrics, confirming the system's consistency and reliability in computing key graph-theoretical features. This not only provided a solid foundation for algorithmic correctness in subsequent analyses but also allowed users to visually understand model behaviour, thereby enhancing the interpretability of the functional correctness validation.

Graph Theory-based Pipeline Validation

Under controlled simulated data conditions, the graph theory-based analysis module underwent a more rigorous functional correctness test. In one representative run (Figure 4.3.c), the software successfully identified all eight predefined significant ROIs within the top eleven most important nodes, a high hit rate of 72.72% during a certain run. This demonstrates the system's ability to capture core topological characteristics of complex networks and to present the results in an interpretable way, reinforcing both transparency and confidence in its functional correctness.

Deep Learning-based Pipeline Validation

Functional correctness was also verified in the deep learning analysis-based module. In a representative run (Figure 4.4.c), the predictions of the MLP regression model showed a strong match with the ground truth (Pearson correlation coefficient $r = 0.90$), with the scatterplot points tightly clustered around the identity line ($y = x$). This indicates that, under controlled conditions, the model could effectively extract predictive signals from the input network and accurately map them to predefined output values, fulfilling the requirement for repeatable functional correctness.

Measurement Example

The functional correctness evaluation in this study employed quantifiable metrics across both pipelines. For the graph theory-based pipeline, verification consisted of checking whether the majority of known ROIs appeared in the predefined "important node ranking," and calculating the hit rate together with confidence intervals. For the deep learning-based pipeline, the assessment focused on the statistical consistency between predicted and actual values using Pearson's correlation coefficient, supplemented by visual inspection of how closely the prediction distribution aligned with theoretical expectations. These measurement methods ensure that functional correctness validation goes beyond qualitative description, providing a standardized basis for cross-experiment and cross-platform comparisons.

5.1.4 Robustness

According to ISO/IEC TS 25058:2024 [17], robustness refers to a software system's ability to maintain stable performance and produce reliable results when facing input perturbations, fluctuations in the operating environment, or noise. In AI/ML-based neuroimaging analysis, robustness is a critical safeguard for ensuring usability under the complex and variable conditions of real-world applications. This attribute requires resilience to multiple sources of uncertainty, including: Variations in data acquisition equipment, noise interference and signal loss, domain shifts in data distribution, and algorithmic randomness from initialization and training processes.

ISO/IEC TS 25058:2024 [17] recognises robustness as a core quality attribute for AI/ML-based systems operating in high-uncertainty environments. The Robustness principle in the FUTURE-AI [26] framework further specifies that performance stability must be preserved despite input perturbations, data variability, or changes in operational conditions, a requirement of particular importance for medical and

neuroimaging AI/ML-based software, where input conditions are often beyond the operator's control.

Graph Theory-based Pipeline Validation

In the graph theory-based analysis module, robustness testing focused on stability under random variations in input data. As shown in Figure 4.3.d, 100 independent runs were conducted using identical parameters and implementation, but with different randomly generated simulated datasets. These network instances shared the same parameter constraints yet had distinct connectivity patterns. In each run, the system was required to identify the top eight most important brain regions out of 20 nodes and verify whether all or most predefined ROIs were included. Results showed that despite random differences in the input data, the hit rate remained consistently high at 94.25%, with a stable distribution. This indicates that, under controlled parameter settings, the pipeline tolerates small structural variations in the input network with minimal fluctuation in outputs, meeting robustness requirements in the "data variability" dimension.

Deep Learning-based Pipeline Validation

In the deep learning-based module, robustness testing assessed the impact of algorithmic randomness on performance. Figure 4.4.d presents results from training the same neural network model 100 times on a fixed dataset. Differences between runs arose solely from the algorithm's internal stochastic processes, such as weight initialisation, mini-batch ordering, and optimisation pathways. The Pearson correlation coefficient remained consistently high across runs (mean $r = 0.8380$, with minimal standard deviation), indicating stable predictive performance despite training randomness. This form of robustness differs from the "data variability" tolerance in the graph theory-based pipeline. Instead, it reflects the model's resilience to uncertainty in the training process, showing that it does not overfit to a specific random initialization or optimization path.

Overall Findings

Together, these robustness tests demonstrate that the implemented AI/ML-based neuroimaging analysis system is tolerant to multiple sources of uncertainty. At the data level, random variations in network structure caused by stochasticity do not lead to substantial deviations in analytical outcomes, while at the algorithmic level, internal randomness from model initialization and training does not compromise predictive stability. This stability not only enhances reliability in the simulation environment but also indicates promising generalization potential for real clinical data, thereby aligning with the FUTURE-AI [26] principles of robustness and transferability.

Measurement Example

The robustness evaluation in this study applied quantifiable and comparable metrics across both pipelines. For the graph theory-based pipeline, robustness was assessed

by recording the distribution of hit rates for predefined ROIs across multiple random input datasets and calculating both mean and variance. For the deep learning-based pipeline, robustness was examined by recording the distribution of Pearson correlation coefficients over repeated training runs with different random initialisations on the same dataset. These approaches ensure that robustness validation is measurable, comparable, and reproducible, thereby providing a standardized reference for assessing robustness across systems and tasks.

5.2 Implications and Generalizations

5.2.1 Implications for Software Research and Testing Methods — Response to Aim 1 (For Developers)

Aim 1 of this study is to provide developers with a reproducible, quantifiable, and systematically extensible quality evaluation framework for AI/ML-based neuroimaging software. This framework is not limited to validating a single tool; it also establishes a methodological foundation for future cross-system comparisons and benchmarking in the field.

Methodological Significance

In terms of research design, this work adopts the experimental simulation pathway from the ABC of software research framework [46], rather than beginning directly with real-world data validation. The reasoning is clear: the high uncertainty inherent in real neuroimaging datasets—such as distribution shifts, inconsistent annotations, and variability in acquisition conditions—makes it difficult to measure software quality attributes precisely. To develop a robust evaluation methodology at an early stage, experiments must first be conducted under fully controllable conditions.

This approach can be illustrated using a "greenhouse" analogy: just as a greenhouse allows precise control over temperature, humidity, and light to isolate the effects of specific variables on plant growth, the contrived simulation environment in this study allows complete control over brain network generation, noise levels, and model parameters [46]. By using the Watts–Strogatz network model with known ground truth, we can isolate and quantify the performance of the three target quality attributes—transparency, functional correctness, and robustness.

Rationale for Choosing Experimental Simulation

The primary reason for adopting an experimental simulation approach lies in its ability to create a fully controlled environment. Within such a setting, the Watts–Strogatz network structure, rewiring probability, and noise levels can be customised, thereby eliminating interference from uncontrollable external factors. This high degree of control enables precise manipulation of experimental variables and ensures that software behaviour can be observed without distortion from unpredictable influ-

ences. Additionally, simulation mitigates the uncertainties inherent in real-world neuroimaging data, such as annotation errors, scanner differences, and other exogenous factors, allowing the evaluation to focus entirely on the performance of the software itself. Finally, the method supports systematic and repeatable testing by establishing stable reference conditions that can be used for future multi-version comparisons and cross-platform benchmarking.

Direct Advantages of This Approach

The approach offers several distinct advantages. First, it ensures reproducibility, as experiments conducted under identical simulation conditions can be rigorously replicated by different research teams, thus enabling verification of results across studies. Second, it facilitates reliability assessment: stability metrics obtained from repeated trials provide a quantitative foundation for evaluating the trustworthiness of the software. Third, it creates benchmarking potential by providing a baseline against which the performance of different AI/ML-based neuroimaging tools can be compared under identical conditions.

Methodological Insights Across Algorithm, Quality, and Compliance Dimensions

From an algorithmic perspective, a controlled simulation environment makes it possible to directly quantify how algorithm behaviour responds to changes in parameters, such as the effect of rewiring probability variations on topological metrics or the influence of different random initialisations on deep learning model performance. Such precise and isolated observations are difficult to achieve using real clinical datasets. In terms of quality assessment, having a known ground truth enables independent and separate measurement of transparency, functional correctness, and robustness, avoiding the confounding effects that often occur when assessing these attributes in complex, real-world settings. From a compliance and standards perspective, this simulation-based testing process aligns closely with the quality attribute measurement definitions in ISO/IEC TS 25058:2024 [17], offering a practical and standardised pathway toward future certification of medical software.

Overall, this framework provides developers with a ready-to-use quality verification method that has the potential to serve as an industry benchmark for neuroimaging software. The logical next step is to expand the evaluation to multi-centre, real-world datasets, thereby enabling a smooth and evidence-based transition from “greenhouse trials” to deployment in clinical environments.

This framework provides developers with an immediately applicable quality verification method that could serve as an industry benchmark for neuroimaging software. The next phase should extend the evaluation to multi-centre real-world datasets, enabling a smooth transition from “greenhouse trials” to clinical deployment.

5.2.2 Implications for Enhancing End-User Accessibility in Quality Testing — Response to Aim 2 (For End-Users)

Complementing the systematic framework in Aim 1 for developers, Aim 2 focuses on neuroimaging software end-users, aiming to provide a simulation-driven, user-friendly testing approach. This method enables researchers without programming backgrounds to independently verify three key quality attributes—Transparency, Functional Correctness, and Robustness—under controlled conditions. Built upon a configurable and visualisable Watts–Strogatz network simulation pipeline, and supported by an intuitive GUI, traceable data management, and repeatable experiment design, the approach lowers the barrier to quality assessment.

Implications for AI/ML-based neuroimaging software testing methods

The simulated data generation method employed in this study is grounded in a mathematically robust model. By combining the configurable Watts–Strogatz pipeline (Figure 4.2) with an interactive GUI (Figures 4.1.a, 4.4.a), the system not only approximates the topological features of real fMRI functional connectivity networks but also preserves mathematical controllability and interpretability—making it suitable for formal verification. Compared to real-world datasets, simulated data eliminates external confounding factors such as scanning artifacts, head motion, and sampling rate variability, allowing precise adjustment of parameters like node count, average degree, and rewiring probability to focus solely on algorithm convergence and system stability. This controlled environment supports rapid iteration and batch evaluation, enabling quick assessment of different graph-theoretical metrics or deep learning hyperparameters, while interactive visualisation helps users detect anomalies and adjust parameters during testing.

Contributions to reproducibility and transparency

The approach also contributes directly to reproducibility and transparency. Multiple test runs (e.g., 100 repetitions) can be performed while maintaining consistent statistical properties, enabling robustness and stability to be quantified through statistical analysis and helping to identify potential instability patterns (Figures 4.3.d, 4.4.d). Saving random seeds and parameter configurations substantially improves reproducibility, creating a reliable basis for benchmarking and peer review. Importantly, the simulation retains realistic network topology while providing known ground truth, allowing for direct validation of functional correctness, robustness, and transparency (Figure 4.2)—a challenge often unmet in real-world data environments. Exported adjacency matrices and metadata (Figure 4.2) ensure complete reproducibility of test scenarios and facilitate direct comparisons across BRAPH 2 versions or alternative tools such as GraphVar or NBS-Predict. Publicly sharing simulation scripts and datasets can reduce cross-team comparison costs, promote community-wide standardisation, and provide a shared reference for quality evaluation in other neuroimaging software.

Contributions to evaluation about software attributes

From an evaluation perspective, the simulation environment offers measurable and controllable conditions for both functional correctness and robustness assessment. Functional correctness can be quantified by computing the hit rate for predefined ROIs (Figure 4.3.c) and by testing theoretical consistency under varying input parameters, such as changes in rewiring probability p (Figure 4.4.c, Pearson $r = 0.90$). Robustness can be examined through boundary testing, such as adjusting network sparsity and randomness to simulate extreme structural scenarios (e.g., sparse, random, or small-world properties) and observing algorithm stability under these conditions (Figures 4.3.d, 4.4.d). For non-deterministic models like deep learning-based pipelines, large-scale repetition (100 runs) can reveal global performance stability under internal randomness—such as weight initialisation and mini-batch ordering—with results showing high average correlation (mean $r = 0.838$) (Figure 4.4.d).

This simulation-driven, user-friendly testing approach aligns closely with the ISO/IEC TS 25058:2024 [17] definitions for measuring transparency, functional correctness, and robustness. It allows end-users to carry out independent quality verification without requiring deep developer involvement, enhancing both the operational feasibility and interpretability of quality assessments. Furthermore, it provides a practical route for building reproducible, shareable, and extensible benchmarking standards, advancing the neuroimaging software community toward standardised quality verification. By aligning with the FUTURE-AI [26] principles of Traceability, Explainability, and Robustness, the approach also holds practical potential for integration into future medical software certification and regulatory review processes.

5.2.3 Generalization of the Software Quality Evaluation Framework

The quality evaluation framework proposed in this study successfully verifies the functional correctness, transparency, and robustness of AI/ML-based neuroimaging software under simulated conditions, demonstrating strong generalizability. Although initial validation was conducted using the BRAPH 2 platform and the Watts–Strogatz network model, the framework is inherently designed to support extension across imaging modalities, analytical tasks, and software platforms.

From a data modeling perspective, the framework is not constrained to a single topological model. While the current study utilizes the Watts–Strogatz model to simulate structural connectivity, the evaluation process is equally applicable to other neuroimaging data modalities, including structural MRI, fMRI, diffusion tensor imaging (DTI), electroencephalography (EEG), and electrocardiography (ECG). The variability in spatial–temporal resolution and noise characteristics across modalities provides a diverse testing ground, supporting the applicability of the framework in heterogeneous data scenarios.

From a task configuration perspective, the framework is not limited to a predefined set of ROIs. Instead, it adopts a flexible ROI specification strategy that does not rely on fixed feature distributions. This design enables the framework to evaluate software quality across a wide range of experimental hypotheses and task set-

tings. As a result, it accommodates varying discriminative brain regions, population heterogeneity, and task complexity—enhancing both interpretability of results and comparability across studies.

From a software perspective, the framework exhibits cross-platform compatibility. Although initially implemented with BRAPH 2, the standardized input-output structure and modular testing logic are designed to support integration with a variety of mainstream neuroimaging platforms, such as FreeSurfer, SPM, GraphVar, and NBS-Predict. By exporting simulated datasets into compatible formats and applying the framework across tools, it becomes possible to perform systematic benchmarking and cross-validation of key quality attributes across platforms.

To facilitate usability and accessibility, the proposed framework has been released as a well-structured distribution package with a user-friendly GUI, available at the GitHub Repository [59], thereby supporting broader generalization across diverse research settings. This distribution enables neuroscience researchers to easily run, visualize, and extend the framework without requiring extensive programming expertise. By lowering technical barriers, it supports reproducibility, promotes wider adoption, and provides an accessible entry point for expanding the framework to new research settings. Beyond this implementation, the framework is not restricted to specific tools or data models but exhibits a high degree of generalizability. It offers a unified foundation for evaluating AI/ML-based neuroimaging software across diverse tasks, modalities, and platforms—thereby advancing trustworthy, reproducible, and platform-agnostic quality assurance practices in computational neuroscience.

5.3 Threats to Validity

"Threats to validity" provide a systematic framework for evaluating the reliability of a study's design and conclusions, thereby enhancing the rigor and credibility of software engineering research. In this study, which focuses on the quality assessment of AI/ML-based neuroimaging software, we have implemented measures such as designing an experimental simulation framework and standardizing quality attribute metrics to partially mitigate validity risks. Nevertheless, further validation using real neuroimaging data and cross-domain applications is necessary to strengthen both the credibility and generalizability of our findings. Following the research methodology framework proposed by Robert [61] and Per [62], the subsequent discussion examines threats to validity from four commonly recognized perspectives.

5.3.1 Construct Validity

Construct validity concerns the correspondence between the theoretical concepts underlying an experiment and the actual observations, namely whether the selected operationalized measures accurately reflect the conceptual constructs in the research question.

1. Rationale for Metric Definition

The testing metrics in this study were designed to evaluate quality attributes such as transparency, functional correctness, and robustness. These metrics were explicitly defined and instantiated with reference to the ISO/IEC TS 25058:2024 [17] standard, the FUTURE-AI [26] framework, and key quality concerns specific to contemporary AI/ML-based neuroimaging analysis software. Furthermore, we proposed actionable measurement methods using the BRAPH 2 software as a case study.

2. Differences Between Simulated and Real-World Data
   Discrepancies in feature distributions between simulated data and real neuroimaging data may cause measurement results to deviate from those observed in actual application contexts. Although we adjusted mathematical model parameters to minimize these differences, further validation with real neuroimaging datasets is required to assess the generalizability of the results.

3. Semantic and Reproducibility Risks
   Differences in the interpretation of quality attribute terminology between researchers and potential end-users (e.g., neuroscientists) may affect the understandability and reproducibility of the proposed methods. In this study, we provided detailed explanations of each attribute's meaning prior to testing, tailored to the experimental needs. Future work could incorporate open-access materials and cross-review by domain experts to ensure consistency and interpretability of definitions.

5.3.2 Internal Validity

Internal validity addresses the reliability of causal inferences, that is, whether the observed results are truly caused by the manipulated variables and whether alternative explanations have been successfully ruled out. The relevant considerations in this study are as follows:

1. Interference from Uncontrolled Variables
   In the simulation experiments, test results may be influenced by uncontrolled factors such as random initialization, hardware variations, or noise levels. To enhance experimental control, we fixed random seeds, standardized the computing environment, and aligned dependency library versions.

2. Causal Link Between Treatment and Outcome
   It is essential to ensure that any differences in evaluation results (e.g., changes in ROI identification accuracy or robustness across repeated runs) truly reflect the system's behavior under the defined test conditions, rather than random external factors. In this study, we adopted a standardized testing procedure based on BRAPH 2 and employed controlled simulated data (Watts–Strogatz networks) to reduce uncontrolled variability. This design increases confidence that the observed outcomes can be reliably interpreted as results of the quality evaluation process, rather than artifacts of uncontrolled influences.

5.3.3 External Validity

External validity concerns the generalizability of research findings. The potential risks and corresponding mitigation strategies in this study are as follows:

1. Limitations of Imaging Modality
   The simulated data pipeline in this study is based on domain-specific features from the fMRI field and the Watts–Strogatz mathematical model, which cannot fully capture the complexity of real neural systems. Future work may extend the approach to other modalities such as DTI and EEG, as well as replicate the experiments using real MRI/fMRI datasets.

2. Generalization to Clinical Data
   Since the current evaluation is primarily based on laboratory conditions and simulated data, it may not fully reflect clinical usage scenarios. Future validation should therefore involve collaboration with medical institutions and the use of actual patient cases to assess clinical applicability.

5.3.4 Conclusion Validity

Conclusion validity addresses the reliability of statistical inferences, including aspects such as sample size, number of repetitions, and significance testing. The main considerations in this study are as follows:

1. Limitations in Diversity of Simulation Conditions
   While simulation data provide the advantage of generating virtually unlimited samples, the current evaluation was performed under a restricted set of conditions, using a single network model configuration (Watts–Strogatz parameters) and fixed testing procedures. This may limit the external validity of the findings, as the framework has not yet been challenged with a broader range of parameter settings, alternative network models, or independent replications by different researchers. To strengthen the reliability and generalizability of the results, future evaluations should systematically vary simulation conditions and incorporate cross-team replications.

2. Potential Subjective Bias in Experiments
   For instance, during the ROI selection phase, variations in the choice of brain regions may introduce subjective differences. To ensure statistical reliability, performance differences should be evaluated through increased repetitions combined with significance testing, and strategies should be adapted to the specific research context (e.g., disease type, regional brain characteristics).

5.4 Usage of Generative AI in This Thesis

The generative AI tool ChatGPT (version 4o and 5) was used to support various aspects of this research. For the main text, it was employed to enhance grammar and academic vocabulary. The tool was not used to generate original content directly. In

the citation section, ChatGPT assisted in formatting references in LaTeX, ensuring consistency and compliance with academic standards. All source materials have been properly cited. For the code implementation, it was used to interpret and annotate existing scripts, enabling a deeper understanding of the software architecture that already had, and it just provided suggestions for debugging improvements, without using generated code.

6

Conclusion

Ensuring the quality of AI/ML-based neuroimaging software remains a critical yet underexplored challenge in neuroscience. As neuroimaging analysis becomes increasingly dependent on complex AI/ML techniques, such as graph theory and deep learning, the issues of transparency, functional correctness, and robustness emerge as key concerns that directly impact reproducibility and clinical applicability. Existing tools often lack systematic quality validation, especially from a software engineering perspective, making it difficult for researchers to assess the reliability of their findings. This thesis aimed to address this gap by proposing a structured, simulation-based evaluation framework that enables rigorous and reproducible software quality assessments; the implemented software quality evaluation framework can be found on GitHub [59].

To tackle this challenge, the study first constructed a highly controllable experimental simulation environment based on the biologically plausible Watts–Strogatz network model. This environment allowed for precise manipulation of network parameters such as average degree, path length, and rewiring probability, providing a reliable benchmark for evaluating algorithm performance. Using this setup, two distinct AI/ML analysis pipelines were tested: a graph theory–based pipeline and a deep learning–based pipeline.

In the graph theory-based pipeline, core topological features were validated against theoretical expectations, and the software demonstrated strong alignment with predefined salient brain regions. The implementation of a GUI further enhanced transparency, allowing non-expert users to track, visualize, and reproduce analytical steps with minimal technical overhead. Functional correctness was confirmed through repeated simulations showing consistent results, while robustness was examined via perturbation experiments that tested the pipeline's stability under varying conditions.

In the deep learning-based pipeline, the system successfully demonstrated through an interactive GUI, users were able to control model configurations and training parameters, ensuring transparency, interpretability, and reproducibility throughout the modeling process. The visualization of simulated brain graphs provided clear insight into data structure and effective region encoding, enhancing interpretability from input to output. Functionally, the model accurately reproduced target outputs,

achieving a high Pearson correlation in a representative training run, thereby confirming functional correctness. Moreover, robustness could be validated through 100 independent training iterations under varying random initializations, with consistently high correlation scores, indicating stable predictive performance despite inherent algorithmic stochasticity. Collectively, these results affirm the testing methodology's capacity to produce reliable and interpretable outcomes in controlled settings, supporting its potential for broader application in AI/ML-based neuroimaging software quality evaluation.

For future work, the framework should be further validated across diverse neuroimaging contexts to ensure its generalizability beyond controlled simulations. While the current pipeline leverages the Watts–Strogatz network model to simulate structural properties of brain connectivity, this mathematical abstraction may not fully capture the complexity of real neuroimaging modalities such as structural MRI, functional MRI, or DTI. Future iterations could repeat the quality assessment process across tasks tailored to different imaging types, and potentially extend the simulation model to other brain-relevant data structures (e.g., EEG, ECG). Additionally, although the proposed methodology is built on standardized, transferable quality attributes, its applicability to widely used AI/ML-based neuroimaging platforms—such as FreeSurfer, SPM, GraphVar, and NBS-Predict—remains to be empirically verified; this could be facilitated by exporting simulated datasets into compatible formats and running the full evaluation pipeline within each environment to generate platform-specific quality reports. As current results are derived solely from simulation, the lack of real clinical imaging data poses a limitation. Collaboration with institutions such as the Karolinska Institute is planned to address this gap and validate the system under practical conditions. Furthermore, to enhance robustness analysis, the framework could include stress testing through synthetic noise injection and the simulation of scanning artifacts to determine tolerance thresholds. Finally, incorporating automated CI/CD pipelines and evaluating a broader set of quality attributes would elevate the framework's maturity for real-world deployment.

In summary, this thesis introduced a systematic, simulation-based approach for evaluating the quality of AI/ML-based neuroimaging software, addressing critical attributes of transparency, functional correctness, and robustness. The dual-pipeline case study, spanning graph theory-based and deep learning-based analyses, demonstrated that controlled simulations can effectively reveal strengths and weaknesses of complex analysis workflows. By combining rigorous software engineering principles with domain-specific evaluation, the proposed framework offers a practical foundation for developing trustworthy, reproducible, and clinically applicable neuroimaging tools, while also providing a scalable pathway for future extensions in both methodology and application scope.

Bibliography

[1] H. Akil, M. E. Martone, and D. C. V. Essen, "Challenges and opportunities in mining neuroscience data," *Science*, vol. 331, no. 6018, p. 708, 2011.

[2] A. Segal, L. Parkes, K. Aquino, S. M. Kia, T. Wolfers, B. Franke, M. Hoogman, C. F. Beckmann, L. T. Westlye, O. A. Andreassen *et al.*, "Regional, circuit and network heterogeneity of brain abnormalities in psychiatric disorders," *Nature Neuroscience*, vol. 26, pp. 1613–1629, 2023.

[3] S. Verdi, A. F. Marquand, J. M. Schott, and J. H. Cole, "Beyond the average patient: How neuroimaging models can address heterogeneity in dementia," *Brain*, vol. 144, no. 10, pp. 2946–2953, 2021.

[4] M. N. Hebart, O. Contier, L. Teichmann, A. H. Rockter, C. Y. Zheng, A. Kidder, A. Corriveau, M. Vaziri-Pashkam, and C. I. Baker, "Things-data, a multimodal collection of large-scale datasets for investigating object representations in human brain and behavior," *eLife*, vol. 12, 2023.

[5] T. Liu, "A few thoughts on brain rois," *Brain Imaging and Behavior*, vol. 5, pp. 189–202, 2011.

[6] W.-S. Sohn, K. Yoo, Y.-B. Lee, S. W. Seo, D. L. Na, and Y. Jeong, "Influence of roi selection on resting state functional connectivity: An individualized approach for resting state fmri analysis," *Frontiers in Neuroscience*, vol. 9, p. 280, 2015.

[7] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, and S. Wagner, "Software engineering for ai-based systems: A survey," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 2, p. Article 37e, May 2021.

[8] C. Gentili, I. A. Cristea, M. Angstadt, H. Klumpp, and K. L. Phan, "The case for preregistering all region of interest (roi) analyses in neuroimaging research," *The European Journal of Neuroscience*, vol. 53, no. 2, pp. 357–361, 2021.

[9] G. Marrelec and P. Fransson, "Assessing the influence of different roi selection strategies on functional connectivity analyses of fmri data acquired during

steady-state conditions," *PLOS ONE*, vol. 6, no. 4, p. e14788, 2011.

[10] M. Mijalkov, L. Storm, B. Zufiria-Gerbolés, D. Veréb, Z. Xu, A. Canal-Garcia, J. Sun, Y.-W. Chang, H. Zhao, E. Gómez-Ruiz, M. Passaretti, S. Garcia-Ptacek, M. Kivipelto, P. Svenningsson, H. Zetterberg, H. Jacobs, K. Lüdge, D. Brunner, B. Mehlig, G. Volpe, and J. B. Pereira, "Computational memory capacity predicts aging and cognitive decline," *Nature Communications*, vol. 16, p. 2748, 2025.

[11] C.-J. Guo, D. Ferreira, K. Fink, E. Westman, and T. Granberg, "Repeatability and reproducibility of freesurfer, fsl-sienax and spm brain volumetric measurements and the effect of lesion filling in multiple sclerosis," *European Radiology*, 2022, in press. [Online]. Available: https://doi.org/10.1007/s00330-018-5710-x

[12] K. Pensionwar Rutuja, M. Anilkumar, and S. Latika, "A systematic study of software quality – the objective of many organizations," *International Journal of Engineering Research and Technology*, vol. 2, no. 5, May 2013. [Online]. Available: https://www.ijert.org/research/a-systematic-study-of-software-quality-the-objective-of-many-organizations-IJERTV2IS50637.pdf

[13] Z. Chen, S. Deng, J. Yin, M. Fu, H. Zhu, L. Yuanping, and T. Xie, "Quality assessment for large-scale industrial software systems: Experience report at alibaba," in *Asia-Pacific Software Engineering Conference (APSEC)*, Dec. 2019. [Online]. Available: https://dblp.uni-trier.de/db/conf/apsec/apsec2019.html#ZhiDYFZLX19

[14] L. Gong, Z. Sun, D. Chen, S. Zhang, and Z. Pang, "Research on testing and quality verification methods of artificial intelligence software," *Science Technology Vision*, vol. 2022, no. 27, pp. 45–48, 2022.

[15] S. Ye, P. Zhang, S. Ji, Q. Dai, T. Yuan, and B. Ren, "A survey of non-functional attributes and quality assurance methods for artificial intelligence software systems," *Journal of Software*, vol. 34, no. 1, pp. 103–129, Jan. 2023.

[16] D. Hou, "Research and application of artificial intelligence software testing," *Electronic Testing*, no. 4, pp. 117–118, 126, 2019.

[17] I. J. S. 42, "Systems and software engineering — systems and software quality requirements and evaluation (square) — guidance for quality evaluation of artificial intelligence (ai) systems," *ISO/IEC TS*, vol. 25058, no. 2024, 2024.

[18] B. Fischl, "FreeSurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, Aug 2012, epub 2012 Jan 10, PMID: 22248573, PMCID: PMC3685476. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2012.01.021

[19] W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols, *Statistical Parametric Mapping: The Analysis of Functional*

Brain Images. London: Academic Press, 2006. [Online]. Available: http://webcat.warwick.ac.uk/record=b2106602~S15

[20] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," *NeuroImage*, vol. 62, no. 2, pp. 782–790, Aug 2012, epub 2011 Sep 16, PMID: 21979382. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2011.09.015

[21] A. Zalesky, A. Fornito, and E. T. Bullmore, "Network-based statistic: Identifying differences in brain networks," *NeuroImage*, vol. 53, no. 4, pp. 1197–1207, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811910008852

[22] E. Serin, N. Vaidya, H. Walter, and J. D. Kruschwitz, "NBS-Predict: An Easy-to-Use Toolbox for Connectome-Based Machine Learning," in *Methods for Analyzing Large Neuroimaging Datasets*, ser. Neuromethods, R. Whelan and H. Lemaître, Eds. New York, NY: Humana, 2025, vol. 218. [Online]. Available: https://doi.org/10.1007/978-1-0716-4260-3_13

[23] M. Mijalkov, E. Kakaei, J. B. Pereira, E. Westman, and G. Volpe, "BRAPH: A graph theory software for the analysis of brain connectivity," *PLOS ONE*, vol. 12, no. 8, p. e0178798, Aug 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0178798

[24] Y.-W. Chang, B. Zufiria-Gerbolés, E. Gómez-Ruiz, A. Canal-García, H. Zhao, M. Mijalkov, J. B. Pereira, and G. Volpe, "BRAPH 2: a flexible, open-source, reproducible, community-oriented, easy-to-use framework for network analyses in neurosciences," *bioRxiv*, 2025, preprint. [Online]. Available: https://doi.org/10.1101/2025.04.11.648455

[25] N. U. Eisty and J. C. Carver, "Testing research software: A survey," *Empirical Software Engineering*, May 2022, published online May 31.

[26] K. Lekadir, A. F. Frangi, A. R. Porras, B. Glocker, C. Cintas, C. P. Langlotz, E. Weicken, F. W. Asselbergs, F. Prior, G. S. Collins *et al.*, "Future-ai: International consensus guideline for trustworthy and deployable artificial intelligence in healthcare," *BMJ*, vol. 388, p. r340, 2025.

[27] H. Liu, K. Li, and Y. Chen, "A review of trustworthiness evaluation and measurement for artificial intelligence systems," *Journal of Software*, vol. 34, no. 8, pp. 3774–3792, 2023, in Chinese.

[28] N. H. Phan, D. Dou, H. Wang, D. Kil, and B. Piniewski, "Ontology-based deep learning for human behavior prediction with explanations in health social networks," *Information sciences*, vol. 384, pp. 298–313, 2017.

[29] C. X. Wang, Q. Wang, and J. G. Li, "Requirements and methods for testing of software as a medical device," *China Med Devices*, vol. 35, no. 11, pp. 66–69,

2020.

[30] X. Cai, H. Lyu, and G. Yu, "A study on the u.s. fda guidelines for reviewing medical ai software," *China Digital Medicine*, vol. 14, no. 11, pp. 34–37, 33, 2019, in Chinese.

[31] H. Wang, S. Li, C. Wang, Q. Tang, J. Li, and J. Li, "Study on special requirements of artificial intelligence medical device quality management system," *China Med Devices*, vol. 36, no. 9, pp. 15–18, 2021.

[32] A. Bowring, C. Maumet, and T. E. Nichols, "Exploring the impact of analysis software on task fmri results," *Human Brain Mapping*, vol. 40, no. 11, pp. 3362–3384, 2018.

[33] J. Zhang and Z. Zhang, "Ethics and governance of trustworthy medical artificial intelligence," *BMC Medical Informatics and Decision Making*, vol. 23, p. 7, 2023. [Online]. Available: https://doi.org/10.1186/s12911-023-02103-9

[34] A. Kiseleva, D. Kotzinos, and P. De Hert, "Transparency of ai in healthcare as a multilayered system of accountabilities: Between legal requirements and technical limitations," *Frontiers in Artificial Intelligence*, vol. 5, p. 879603, 2022.

[35] E. Haddad, F. Pizzagalli, A. H. Zhu, R. R. Bhatt, T. Islam, I. Ba Gari, D. Dixon, S. I. Thomopoulos, P. M. Thompson, and N. Jahanshad, "Multisite test-retest reliability and compatibility of brain metrics derived from freesurfer versions 7.1, 6.0, and 5.3," *bioRxiv*, April 2022. [Online]. Available: https://www.biorxiv.org/content/biorxiv/early/2022/04/14/2022.04.13.488251.full.pdf

[36] L. Clerx, E. H. B. M. Gronenschild, C. Echavarri, F. R. J. Verhey, P. Aalten, and H. I. L. Jacobs, "Can freesurfer compete with manual volumetric measurements in alzheimer's disease?" *Current Alzheimer Research*, vol. 12, no. 4, pp. 358–367, 2015.

[37] Y. Chatelain, L. Tetrel, C. J. Markiewicz, M. Goncalves, G. Kiar, O. Estéban, P. Bellec, and T. Glatard, "A numerical variability approach to results stability tests and its application to neuroimaging," *IEEE Transactions on Computers*, January 2024.

[38] Y. Chen, H. Li, D. Hu, H. Qi, and H. Chen, "Design of software for multimodal radiomics feature mining and analysis based on artificial intelligence," *Chinese Journal of Medical Physics*, no. 12, pp. 1578–1584, 2024, in Chinese.

[39] H. Zhang, "Development and application of brain imaging data analysis system based on docker containerization technology," Master's thesis, University of Electronic Science and Technology of China, 2022, in Chinese.

[40] H. Lee, S. Tajmir, J. Lee, M. Zissen, B. A. Yeshiwas, T. K. Alkasab, G. Choy, and S. Do, "Fully automated deep learning system for bone age assessment," *Journal of Digital Imaging*, vol. 30, no. 4, pp. 427–441, 2017.

[41] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[42] O. Abdelaziz, C. Spampinato, S. Palazzo *et al.*, "Automatic brain tumor segmentation based on cascaded convolutional neural networks with uncertainty estimation," *Frontiers in Computational Neuroscience*, vol. 13, p. 56, 2019.

[43] S. M. Plis, A. D. Sarwate, D. Wood, C. Dieringer, D. Landis, C. Reed, S. R. Panta, J. A. Turner, J. M. Shoemaker, K. W. Carter, P. Thompson, K. Hutchison, and V. D. Calhoun, "Coinstac: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data," *Frontiers in neuroscience*, vol. 10, p. 365, 2016.

[44] T. Glatard, L. B. Lewis, R. Ferreira da Silva, R. Adalat, N. Beck, C. Lepage, P. Rioux, M.-E. Rousseau, T. Sherif, E. Deelman *et al.*, "Reproducibility of neuroimaging analyses across operating systems," *Frontiers in Neuroinformatics*, vol. 9, p. 12, 2015.

[45] Z. Ye, "Application of artificial intelligence technology in software testing," *Shanghai Electric Technology*, vol. 17, no. 2, pp. 78–81, 2024, in Chinese.

[46] K.-J. Stol and B. Fitzgerald, "The abc of software engineering research," *ACM Transactions on Software Engineering and Methodology*, vol. 27, no. 3, pp. 11:1–11:34, Sep 2018.

[47] J. E. Chen and G. H. Glover, "Functional magnetic resonance imaging methods," *Neuropsychology review*, vol. 25, no. 3, pp. 289–313, 2015.

[48] E. Bullmore and O. Sporns, "Complex brain networks: Graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, pp. 186–198, 2009.

[49] M. G. Hart, R. J. F. Ypma, R. Romero-Garcia, S. J. Price, and J. Suckling, "Connectome analysis for pre-operative brain mapping in neurosurgery," *British Journal of Neurosurgery*, vol. 30, no. 5, pp. 506–517, 2016.

[50] K. J. Friston, "Functional and effective connectivity: a review," *Brain connectivity*, vol. 1, no. 1, pp. 13–36, 2011.

[51] M. Cao, N. Shu, Q. Cao, Y. Wang, and Y. He, "Imaging functional and structural brain connectomics in attention-deficit/hyperactivity disorder," *Molecular Neurobiology*, vol. 50, no. 3, pp. 1111–1123, Dec 2014, epub 2014 Apr 5. Review.

[52] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[53] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[54] Q. Cao, N. Shu, L. An, P. Wang, L. Sun, M.-R. Xia, J.-H. Wang, G.-L. Gong, Y.-F. Zang, Y.-F. Wang, and Y. He, "Probabilistic diffusion tractography and graph theory analysis reveal abnormal white matter structural connectivity networks in drug-naive boys with attention deficit/hyperactivity disorder," *The Journal of Neuroscience*, vol. 33, no. 26, pp. 10 676–10 687, Jun 2013.

[55] L. Wang, C. Zhu, Y. He, Y. Zang, Q. Cao, H. Zhang, Q. Zhong, and Y. Wang, "Altered small-world brain functional networks in children with attention-deficit/hyperactivity disorder," *Human Brain Mapping*, vol. 30, no. 2, pp. 638–649, 2009.

[56] M. P. van den Heuvel, C. J. Stam, R. S. Kahn, and H. E. Hulshoff Pol, "Efficiency of Functional Brain Networks and Intellectual Performance," *The Journal of Neuroscience*, vol. 29, no. 23, pp. 7619–7624, 2009. [Online]. Available: https://doi.org/10.1523/JNEUROSCI.1443-09.2009

[57] A. J. Karran, T. Demazure, A. Hudon, S. Senecal, and P.-M. Léger, "Designing for confidence: The impact of visualizing artificial intelligence decisions," *Frontiers in Neuroscience*, vol. Volume 16 - 2022, 2022. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.883385

[58] A. Burns, C. Lee, T. On, C. Xiong, E. Peck, and N. Mahyar, "From invisible to visible: Impacts of metadata in communicative data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 7, pp. 3427–3443, 2024.

[59] Y. Guo, Y.-W. Chang, and G. Volpe, "Wattsstrogatzmodel: This is the braph 2 watts–strogatz model distribution that tailors the simulation of realistic fmri data." [Online]. Available: https://github.com/yu-wei-c/WattsStrogatzModel

[60] A. Argun, A. Callegari, and G. Volpe, *Simulation of Complex Systems*. IOP Publishing, 2021, pp.12-8. [Online]. Available: https://dx.doi.org/10.1088/978-0-7503-3843-1

[61] R. Feldt and A. Magazinius, "Validity threats in empirical software engineering research – an initial survey," in *Proceedings of the International Conference on Software Engineering and Advanced Applications (SEAA)*. Gothenburg, Sweden: IEEE, Jan 2010, pp. 374–379.

[62] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, Apr 2009.