

From Theory to Practice:
The Transformation of Computer Science through Experimental Research

Author: Munteanu Bianca-Ştefania

Abstract

This paper delves into the evolving landscape of experimental research methods in computer science, with a focus on their impact, challenges, and future prospects. It begins by tracing the evolution of experimental methods from intuition-based approaches to systematic empirical methodologies, highlighting their role in significant advancements in AI, software systems, and algorithm design. The core of the paper addresses the challenges experimental methods face, such as issues of reproducibility, and the complexity of practical application.

Further, the paper explores the instrumental role of these methods in advancing AI applications in healthcare and education, citing specific examples such as AI-driven cancer detection and the development of educational robots. Looking ahead, the paper discusses the vast potential of experimental research in computer science, emphasizing the need for robust, innovative methods to enhance ethical frameworks, improve data diversity, and bridge the gap between theoretical and practical applications. This comprehensive analysis aims to provide a deeper understanding of the significance of experimental research in computer science and its indispensable role in the future of technological innovation.

Chapter 1. Introduction

Computer Science represents the science of computers and algorithms, being a discipline that uniquely combines abstract theoretical elements with practical ones. This has its foundations in mathematics, but it differs from it precisely by focusing on practical elements. With the emergence of the concept that programs are stored in memory during the 1940s, computer science began to develop as an independent discipline, gradually separating itself from mathematics and electrical engineering. This transition was accompanied by intense debates in the 1960s and 1970s regarding the mathematical fundamental characteristics and principles underlying the field of computer science and the role of experimentation in it. These debates included diverse views of computer science, from those who focused on the detailed mathematical examination of algorithms to derive broad conclusions, to those who promoted a theoretical framework based on mathematics for building software programs (Matti Tedre, 2014).

Although computer science is strongly interdisciplinary, its classification as a "science" has been and continues to be a matter of debate. Some researchers argue that computer scientists should rather be considered engineers or programmers, given the applied nature of the field (Frédéric Desprez, 2012). Throughout its evolution, computer science has witnessed a continuous debate about its scientific character. The debates included questions such as whether computer science is a science and, if so, what exactly it studies - from computers and the phenomena around them, encompassing a range of areas including data and information processing, algorithm development, various computation categories, the art of programming, intricacies of computational complexity, and procedural methodologies. These debates failed to definitively place computer science in a concrete scientific framework, but they shifted attention from subjects to activities, arguing that computer science is a scientific discipline not through its subject but by its method of inquiry.

In computer science, research is a systematic, organized process that aims to establish facts, reach new conclusions, and test hypotheses. While general definitions of research highlight the systematic investigation of materials to generate new knowledge, specific fields such as computer science adapt

this definition to their unique contexts. In computer science, research involves not only the empirical collection of information, but also the development and testing of new algorithms, software, and systems. It is a process characterized by formulating specific research questions, applying methodical approaches to address these questions, and presenting the results obtained through these methods. This adaptation of the definition of research in computer science emphasizes the discipline's focus on logical, confirmable, and practical research, especially in areas such as algorithm development, software engineering, and artificial intelligence (Hassani, 2017).

The objective of this paper is to deepen the role of experimental methodology in the rapid progress of computer science. It aims to explore how experimentation has influenced different subfields of computer science and how it continues to shape the future of technological innovation. Also, the paper addresses the following research questions:

- RQ1. How have experimental research methods evolved in computer science, and what impact have they had on the field?
- RQ2. What are the main challenges and limitations associated with experimental research in computer science?
- RQ3. How do experimental methods in computer science compare with traditional research methods in terms of effectiveness and results?
- RQ4. What role do experimental research methods play in the advancement of artificial intelligence, particularly in domains like healthcare and education?
- RQ5. What are the future prospects and potential directions for experimental research in computer science?

Chapter 2. Fundamentals of experimental research method

2.1. Key Concepts

In exploring the fundamentals of experimental research methods in computer science, it is essential to delve into the core concepts that define this domain. As highlighted in the insightful article (Frédéric Desprez, 2012) experimental methodology within computer science is not just a procedural aspect, but a cornerstone that underpins the entire research process. This subchapter aims to elucidate these key concepts, providing a foundation for understanding the significance of experimental research in the field.

2.1.1. Quantitative vs. Qualitative Research

- **Quantitative Research:** Involves collecting and analyzing numerical data to understand patterns, relationships, and causality. In computer science, this might include experiments that measure performance metrics like processing speed, memory usage, or error rates.
- **Qualitative Research:** Focuses on understanding concepts, experiences, or phenomena through non-numerical data. In the context of computer science, it might involve case studies, interviews, or content analysis to explore user experiences, software usability, or the social impact of technology.

2.1.2 Hypothesis

A hypothesis in experimental research is a tentative statement predicting a potential outcome or explaining a phenomenon. It's formulated based on existing knowledge and observations and guides the direction of the study. In computer science experiments, a hypothesis might predict how a certain algorithm will perform under different conditions.

2.1.3. Variables

- **Independent Variables:** These are variables that the researcher changes or manipulates in an experiment. In computer science, this could be the type of algorithm or hardware configuration used.
- **Dependent Variables:** These are the variables that are observed and measured to see the effect of the independent variable. For example, in a computer science experiment, this might be the time taken to complete a task.
- **Control Variables:** These are the variables that remain unchanged in the experiment to ensure that any observed effects on the dependent variable are exclusively attributed to the variation in the independent variable, rather than external factors.

2.1.4. Experimental Design

This refers to the overall strategy used to conduct an experiment. It includes the methodology for selecting participants, allocating them to groups, applying interventions, and measuring outcomes. In computer science, experimental designs might include controlled laboratory experiments, field experiments, or simulations.

2.1.5. Sampling

In research, sampling denotes the method of choosing a representative segment from a broader group to be involved in a study. The goal is to draw conclusions that are generalizable to the larger population. In computer science, sampling might involve selecting a representative set of data points, users, or systems to study.

2.1.6. Experimental Methodology

This encompasses the comprehensive set of procedures used to conduct an experiment. It includes the steps from initial planning to the final analysis and reporting of results. In computer science, this might involve defining experimental conditions, selecting appropriate metrics, and using tools for data analysis and visualization.

2.1.7. Basic Services in Experimental Research

- **Resource Management:** This involves the allocation and management of resources such as computing power, storage, and network capabilities necessary for conducting experiments.
- **Data Management:** Refers to the practices of collecting, storing, and using data efficiently, securely, and ethically in experiments.
- **Instrumentation and Control** Involves the techniques and tools used to monitor and control the variables in an experiment. This can include software and hardware tools for automating tasks, collecting data, and ensuring the experiment runs as intended.

2.2. History of experimental research

The article (Moher, 1982) provides a succinct overview of the historical progression of experimental research in computer science, tracing its evolution from a discipline guided by intuition and common sense to one that increasingly embraces empirical and experimental methods.

- **The Early Days: Intuition and Common Sense (Pre-1970s)**

In the nascent phase of software engineering, the field was primarily driven by personal insights, intuition, and what was commonly accepted as practical wisdom. Instead of relying on controlled experiments or scientific principles, early software engineering was shaped by a collection of

guidelines and „rules of thumb” developed through practical experiences. For instance, concepts like top-down modular programming gained traction not because of empirical evidence but due to their apparent practical benefits. During this period, debates in the field, such as those over the use of GOTO statements or structured programming, were often based on subjective experiences rather than scientific validation.

- The Advent of Experimental Studies (Late 1960s to Early 1970s)

The shift towards a more experimental approach in computer science began in the late 1960s. This period saw the first systematic studies examining aspects of the programming process, such as the comparison between time-sharing and batch-computing environments. Although these early experiments didn't provide conclusive results, they marked a significant shift towards a more empirical approach in the field. The publication of Gerald Weinberg's "The Psychology of Computer Programming" in 1971 was pivotal in this transition. Weinberg's book approached software development as a complex human activity amenable to experimental study, sparking interest in testing the principles of software engineering through controlled experiments. The period between 1972 and 1974 saw the emergence of various studies, such as those examining programming language constructs and the effects of coding style on software development.

- Expansion and Diversification (Post-1975)

From 1975 onwards, there was a significant surge in the use of experimental methods to study various aspects of computer science, particularly programming languages and human-computer interaction. This era saw a broadening of research topics, ranging from the design and comprehension of programming languages to the intricacies of the software development process. Notable studies during this period include those evaluating the reliability of different programming languages and the usability of database query languages, particularly for non-programmers. This period also witnessed the development of foundational theories in programming and attempts to create frameworks for understanding the cognitive processes involved in programming. The research in this era was characterized by a move towards confirming or refuting the claims made by earlier practitioners, marking a significant shift towards an empirical, evidence-based approach in software engineering and language design.

2.3. Comparison with other research methodologies

The field of computer science, emerging from its roots in mathematics and engineering, has often been seen as an empirical discipline. It blends theoretical aspects like formal models and abstraction with practical applications in software engineering. This duality is reflected in its research methods, which can be both theoretical and experimental. Unlike other sciences where experimental methods might strictly adhere to certain paradigms, research in computing often employs a mix of quantitative and qualitative methods. This flexibility allows computing research to adapt methods from other well-established sciences while addressing its unique challenges and identity issues. As the field continues to evolve, the importance of a well-defined research framework that accommodates the interdisciplinary nature of computing becomes increasingly vital (Hassani, 2017).

Research methodologies can also be categorized based on their application and objectives. Pure research, for instance, seeks to generate new knowledge without immediate application, while applied research aims to solve specific, current problems. Descriptive research focuses on explaining specific problems to inform further research. In contrast, exploratory research seeks new information in unexplored areas to develop hypotheses. Correlational research investigates relationships between variables, and explanatory research delves into understanding the reasons behind phenomena. Analytical research goes a step further, aiming to uncover the reasons behind a problem or phenomenon.

Chapter 3. Conducting effective experimental research in computer science

In the dynamic field of computer science research, the journey from initial inquiry to conclusive findings is shaped by a structured and meticulous methodology. The preceding chapter serves as a comprehensive guide through this journey, elucidating the fundamental steps of successful research methodology.

3.1. Research questions

The nature of questions in computer science is diverse, covering theoretical, empirical, and experimental aspects. Questions often arise from practical problems, technological advancements, or theoretical explorations. Unlike more established disciplines, computing research does not have globally accepted methods. Each research project in computing often requires a specific adjustment of general research methods to suit its unique challenges and objectives (Hassani, 2017).

Given the interdisciplinary nature of computer science, research questions often incorporate methods and perspectives from fields like mathematics, engineering, or social sciences. This approach not only enriches the research but also makes it more comprehensive in addressing complex, real-world problems. The rapid pace of technological change in the field further influences these questions, requiring researchers to be flexible and forward-thinking in their approach. As such, the questions must not only address current technological challenges but also anticipate future trends and developments.

Furthermore, the collaboration between academia and industry in computer science encourages the development of research questions that bridge theoretical research and practical application. This synergy ensures that the research is not only academically rigorous but also has tangible real-world applications, driving innovation in the field. The influence of funding agencies and industry partners can also play a significant role in shaping these questions, as they often set priorities and focus areas for research.

3.2. Hypothesis

Expanding on the concept of hypothesis in computer science research, it's crucial to distinguish between two different levels at which hypotheses operate: the macro level and the micro level. This distinction is particularly relevant in the context of experimental computer science, as described in (Feitelson, 2006).

Macro-level hypotheses concern the underlying principles that shape an entire field of study. These hypotheses are not systematically proposed and tested but rather emerge as summarizing principles uniting extensive bodies of work. For example, the P vs NP problem is a fundamental hypothesis in theoretical computer science that suggests that no efficient polynomial algorithms exist for NP-complete problems. This hypothesis has guided significant research in algorithm development and problem classification.

In contrast, micro-level hypotheses are tailored for specific experiments. They focus on generating predictions about certain aspects of computer systems. The process typically involves analyzing a computer system to gather data, followed by modeling this data to elucidate the system's behavior. The hypothesis is then subjected to tests against additional data, aiming to confirm that the model accurately depicts the system's inner workings.

3.3 Experiment

3.3.1. Types of experiments

Over time, the approaches and goals of experimental research in computer science have been continuously changed, a fact that led to the grouping of experiments into 5 important categories, as described in the article (Matti Tedre, 2014).

- **Feasibility experiments:** In this category, we include demonstrations or experiments considered "proof of concept". They aim to show that a certain task, for which the efficiency and the real possibility of automation are not known, can actually be executed successfully. This approach often involves developing new technologies or systems to establish their feasibility. Their purpose is to transform theory into practical demonstration, focusing on proving existence, and not necessarily on performing a rigorous experimental analysis.
- **Trial experiments:** Going beyond simply demonstrating feasibility, trial experiments evaluate various aspects of a system under a predetermined set of variables. These experiments are designed to test how well a new system meets its specifications or performs under certain conditions. They are generally carried out in virtual environments and include methods such as emulation, simulation, and benchmarking. They are crucial to both applied and theoretical aspects of computer science and are often used to validate computational models and systems.
- **Field experiments:** They aim to evaluate aspects similar to trial experiments, but test the performance of a system in the real environment, with real users. These experiments study how a system meets its purpose and requirements in the context for which it was designed, evaluating factors such as performance, usability, and robustness. Field experiments offer a more realistic evaluation compared to studies in virtual environments, although they are less controllable and very difficult to reproduce.
- **Comparison experiments:** These are based on the comparison of different solutions or systems to determine the most efficient one, based on certain criteria. Although comparison experiments are intended to be objective, they are susceptible to researcher-induced biases. Standardized tests and benchmarking are often used to prevent the problem and provide a fair comparison.
- **Controlled experiments:** Considered the gold standard in scientific research, controlled experiments aim to eliminate confounding factors to allow generalization and prediction. These experiments are used to determine if there is an association between variables or if one variable causes an effect on another. Controlled experiments are particularly valued for their ability to provide probabilistic knowledge about causality and are widely used in fields such as software engineering and computational security, advocating for more generalizable and justifiable results about software products.

3.3.2 Importance of reproducibility

In the field of computer science, the reproducibility of an experiment is a fundamental aspect that gives transparency and credibility to the results obtained. This refers to the ability of independent researchers to replicate the results of a study using the same methods and data as in the original research. In order to achieve this, there must be some rigorous standards and practices in the development of research documentation.

The article (Kjensmo, 2017) emphasizes the importance of reproducibility, assessing the extent to which published research in computer science meets specific reproducibility criteria, such as the availability and documentation of experimental procedures, methods, and data. In principle, two important aspects are followed: the reproducibility of the methods and the reproducibility of the results.

- **Reproducibility of methods:** This involves disclosing enough information so that another researcher could theoretically perform the same procedures with the same data. This involves detailed documentation of the experiment, methods, and data used in the research.
- **Reproducibility of results:** This aspect focuses on the ability of an independent researcher to corroborate results using the same experimental procedures. It does not necessarily require the original data set but emphasizes that the corroborated results should align with the original findings.

In addition to these two key aspects, reproducibility also involves accuracy and clarity in documentation. It is essential that methodological details are accurately described, including software versions, configuration settings, and algorithms used. Clarification of processes is also important to enable step-by-step replication of the experiment.

3.4. Modeling and Analysis

In computer science, the analysis of data, particularly through modeling, is a pivotal aspect of research. This process involves several stages, from collecting data to extracting meaningful insights.

Quantitative research (Apuke, 2017) in computer science often involves gathering numerical data and applying statistical methods to analyze these variables. This analysis aims to answer specific questions or test hypotheses, utilizing mathematical and statistical techniques. In quantitative research, data are collected and then analyzed using statistical methods. This approach is essential for validating hypotheses and exploring cause-and-effect relationships in a measurable and objective way.

Modeling in data analysis refers to the creation of abstract representations of systems based on the collected data. It's a method of simulating, understanding, and predicting system behavior by constructing mathematical or computational models (Tichy, 1998).

Objectivity is a critical aspect of data analysis in computer science. Researchers aim to minimize their biases and ensure that the data analysis is as unbiased and accurate as possible. This objective stance helps in drawing reliable and valid conclusions from the data (Apuke, 2017).

3.5. Conclusion

The conclusion step in this scientific method, particularly in the context of computer science research, is a critical phase where the results of experiments and analyses are interpreted. This is not just about analyzing the end results, it also involves considering the entire process of observation and measurement that led to these results. This comprehensive view ensures that the conclusions drawn are well-grounded in empirical evidence and methodological rigor. This involves exploring the outcomes and understanding the implications of these findings in relation to the original hypotheses and research questions. As mentioned by (Matti Tedre, 2014) the main objectives of the conclusion step are:

- **Confirming or Refuting Hypotheses:** One primary objective at this stage is to determine whether the results support or contradict the initial hypotheses. This process may validate existing theories or indicate the need for their revision.
- **Generating New Insights:** The conclusion phase often leads to new insights, which can be either confirmatory, offering support to existing knowledge, or revelatory, uncovering new aspects previously unconsidered.

- **Refinement and Development:** Based on the conclusions drawn, existing theories might be refined, or new research questions and hypotheses may be developed for further exploration. The insights and conclusions drawn from one study can significantly inform future research directions.

Chapter 4. Examples of revolutionary experiments in artificial intelligence

In the rapidly evolving landscape of computer science, experimental methodology plays a crucial role in advancing our understanding and application of emerging technologies. This chapter delves into the realm of experimental research within computer science, focusing on cutting-edge applications in artificial intelligence. By meticulously analyzing outcomes and iterating processes, these experiments contribute significantly to the field's advancement, propelling innovations from theoretical concepts to real-world applications. This chapter aims to provide a comprehensive overview of these experimental endeavors, showcasing how they are instrumental in shaping the future of technology and its application in our daily lives.

4.1 GPT-4

The Article (Bubeck, 2023) presents a comprehensive overview of the diverse applications of experimental methods in the field of computer science, with a specific focus on AI research. The study centers around GPT-4, a sophisticated large language model from OpenAI. This model demonstrates an impressive array of capabilities across various domains, challenging our conventional understanding of learning and cognition.

The experiments conducted with GPT-4 uncover several key areas where experimental research methods are pivotal within computer science. In the realm of Multimodal and Interdisciplinary Composition, GPT-4 exhibits remarkable integrative abilities across domains such as vision, image generation, and music. This reflects the model's ability to transcend mere language proficiency, tackling complex tasks across diverse fields and offering insights into the integrative and interdisciplinary nature of AI research.

When it comes to Coding, GPT-4's proficiency in translating instructions into code, addressing coding challenges, and understanding existing code highlights the crucial role of experimental methods in both real-world scenarios and theoretical problem-solving. This underscores the model's versatility and adaptability in the face of diverse computational challenges.

In assessing Mathematical Abilities, GPT-4's interactions with mathematical problems—ranging from casual conversations about mathematics to performance on specific problem datasets—serve as a testament to the utility of experimental methods in evaluating and enhancing the cognitive abilities of AI in abstract domains.

The area of Interaction with the World further showcases GPT-4's capabilities, particularly through experiments in tool use and embodied interaction. These include solving complex tasks using multiple tools and navigating through various environments, thereby illuminating the effectiveness of experimental methods in appraising AI's practical applications and real-world interactions.

Moreover, the Interaction with Humans segment of the study delves into GPT-4's experiments in understanding human theory of mind and enhancing explainability. This illustrates the depth to which experimental methods in AI can explore the nuances of human-AI interaction and communication.

In exploring Discriminative Capabilities, the model's proficiency in detecting personally identifiable information, fact-checking, and identifying misconceptions highlights the role of experimental methods in bolstering AI's discriminative capabilities. This is particularly relevant for fostering responsible and ethical AI development.

The Societal Influences section of the research contemplates the broader implications of AI advancements. It discusses challenges such as misinformation, manipulation, and bias, emphasizing the significant impact of experimental methods in understanding and addressing the societal effects of technological advancements.

Lastly, the study concludes with Future Directions, reflecting on the definitions of intelligence, AI, and AGI, and contemplating the path towards a more general form of artificial intelligence. This highlights the evolving nature of experimental methods in influencing the future trajectory of AI research and development.

4.2. Healthcare AI

The experimental approach in the study (Patel, 2020) involves the application of AI techniques to specific types of cancer, namely breast, lung, and liver cancers. These AI experiments are crucial for several reasons, each contributing to the advancement of cancer diagnosis and treatment.

The first and foremost aspect of these experiments is the validation of various AI techniques. Experiments utilizing neural networks, for example, demonstrate their capacity to identify early-stage lung cancer cells. This is particularly vital considering lung cancer's high mortality rate. Such experiments not only confirm the effectiveness of these techniques but also highlight AI's potential in identifying cancers at a stage where they are most treatable.

Moreover, these experiments involve a thorough comparison and optimization of different AI models. This comparative analysis is essential in discerning the most efficient model for cancer detection. In the case of liver cancer, for instance, the juxtaposition of methods like particle swarm optimization, neural network classifiers, and deep learning models reveals different accuracy levels. These findings underscore the potential for further refining these methods and enhancing their diagnostic precision.

The experiments also introduce innovative approaches to cancer detection. In breast cancer detection, for instance, the application of AI techniques aims to improve early detection rates, a crucial factor in successful treatment. The deployment of advanced methodologies like deep learning in these experiments reflects the evolving nature of AI applications in healthcare, pushing the boundaries of traditional diagnostic methods.

Lastly, the focus on data management and processing in these AI applications is a significant aspect of these experiments. Managing large datasets, along with processes like image preprocessing, feature extraction, and the optimization of neural networks, forms the backbone of these experimental methodologies. This focus emphasizes the complexity of effectively handling data in AI-driven cancer detection, an essential component in the accuracy and reliability of these AI systems.

In essence, the experiments conducted in the study are foundational in testing and proving the viability of AI applications in cancer detection. They offer insights into the most effective techniques, paving the way for innovative approaches in AI-driven healthcare, and address the challenges of data management and processing. The experimental results are a testament to the potential of AI in revolutionizing cancer diagnosis, making early detection more accurate and efficient, thereby potentially saving countless lives.

4.3. Educational AI

The article (Timms, 2016) digs into the integration of Artificial Intelligence in Education (AIED) with advancements in robotics, aiming to revolutionize traditional teaching and learning environments. „Educational Cobots” are collaborative robots designed to assist human teachers in classrooms. These cobots, leveraging the capabilities of AIED, are envisioned to play a vital role in

maintaining student engagement, providing tailored instruction, and assisting in classroom management.

The rationality behind embedding AIED into robotic systems is rooted in the concept of embodied cognition, which posits that human intelligence is intertwined with our physical bodies and sensorimotor systems. This perspective suggests that for AI systems to effectively mimic human intelligence, they need to be able to navigate and interact within physical environments, much like humans do. Educational cobots, therefore, are designed to move around classrooms, recognize individuals, and use standard classroom equipment, thereby enhancing their interaction with students and teachers.

One of the key experiments highlighting the effectiveness of educational cobots involved a robot named Rubi, which was used in a study at the Temporal Dynamics of Learning Centre. Rubi was programmed to respond to the gaze of young children, resulting in a significant increase in their attention span. This experiment underscores the potential of robots to engage students in ways that traditional educational tools might not. In Japan, a humanoid robot named Saya was used in classrooms, demonstrating the possibility of robots performing basic instructional tasks and interacting socially with students.

Additionally, experiments with the Robovie social robot in elementary science classrooms explored whether robots could stimulate students' curiosity in science. While the overall curiosity among students didn't increase, those who interacted with Robovie showed increased interest, hinting at the potential of robots to engage students in specific contexts.

The movement towards educational cobots has seen countries like Japan and South Korea exploring the use of more lifelike robots in education. This interdisciplinary effort requires collaboration between AIED researchers and roboticists to create effective educational technologies. Such collaboration necessitates shared knowledge across disciplines like computer science, cognitive psychology, instructional design, and psychometrics.

To further this vision, the use of intermediary technologies such as natural language processing and tactile interfaces is essential. These technologies facilitate more naturalistic interactions between humans and robots, enhancing the educational experience. Advances in virtual and augmented reality, like the Oculus Rift, also hold potential for application in smart classrooms and with educational cobots.

In conclusion, the experiments and developments discussed in the article illustrate the profound impact that integrating AI and robotics can have on education. Educational cobots, equipped with AI capabilities, have the potential to transform traditional learning environments, making education more interactive, engaging, and effective.

Chapter 5. Conclusion

In this final chapter, we'll look back at the important questions we asked at the beginning of this paper. Our goal is to summarize what we've learned about experimental research methods in computer science. We'll talk about how these methods have changed the field, the challenges they face, how they compare to other methods, and what they could mean for the future. Let's revisit these questions and see what we've discovered.

- RQ1. Evolution and Impact of Experimental Methods in Computer Science

The analysis of experimental research methods in computer science reveals a significant evolution from intuition-based approaches to systematic empirical methodologies. These methods have enabled groundbreaking developments in AI, software systems, and algorithm design. The impact of such

research is evident in the advancements in multimodal AI systems like GPT-4, which have revolutionized how we interact with technology and process vast amounts of information.

- RQ2. Challenges and Limitations

Despite their successes, experimental research methods in computer science face challenges, including issues of reproducibility, ethical considerations, and the complexity of translating theoretical models into practical applications. The limitations in data availability and the bias in AI algorithms also pose significant hurdles that need to be addressed to ensure the validity and reliability of research outcomes.

- RQ3. Comparison with Traditional Methods

Comparatively, experimental research methods have shown greater effectiveness in yielding tangible, real-world applications than purely theoretical approaches. These methods provide a more dynamic and responsive framework for testing and refining computer science theories, especially in interdisciplinary contexts.

- RQ4. Role in Advancing AI in Healthcare and Education

In domains like healthcare and education, experimental research methods have been instrumental. In healthcare, AI-driven experimental research has led to more accurate cancer detection methods, while in education, it has facilitated the development of educational robots, enhancing learning experiences.

- RQ5. Future Prospects and Directions

Looking ahead, the potential for experimental research in computer science is vast. With the increasing integration of AI and machine learning in various sectors, the need for robust, innovative experimental methods is more critical than ever. Future research should focus on enhancing the ethical framework, improving data diversity, and bridging the gap between theoretical research and practical applications.

Bibliografie

- Apuke, O. (2017). Quantitative Research Methods : A Synopsis Approach. *Arabian Journal of Business and Management Review (kuwait Chapter)*, 6(10). doi:10.12816/0040336
- Bubeck, S. a. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Feitelson, D. G. (2006). Experimental computer science: The need for a cultural change. *Internet version: <http://www.cs.huji.ac.il/~feit/papers/exp05.pdf>*, 1--37.
- Frédéric Desprez, G. F. (2012). Supporting Experimental Computer Science. Preuat de pe <https://inria.hal.science/hal-00720815>
- Hassani, H. (2017, 03). Research Methods in Computer Science: The Challenges and Issues. *arxiv.org*.
- Kjensmo, S. (2017). *Research method in AI-Reproducibility of results*.
- Matti Tedre, N. M. (2014). "Experiments in Computing: A Survey". *The Scientific World Journal*, 11. doi:<https://doi.org/10.1155/2014/549398>
- Moher, T. a. (1982). Methodology and experimental research in software engineering. *International Journal of Man-Machine Studies*, 16(1), 65--87.
- Patel, D. a. (2020). Implementation of artificial intelligence techniques for cancer detection. *Augmented Human Research*, 5, 1--10.
- Tichy, W. (1998). Should computer scientists experiment more? *Computer*, 31(5), 32-40. doi:10.1109/2.675631
- Timms, M. J. (2016). Letting artificial intelligence in education out of the box: educational cobots and smart classrooms. (2016, Ed.) *International Journal of Artificial Intelligence in Education*, 26, 701--712.