ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

DESIGN AND USABILITY EVALUATION OF AN END-USER PROGRAMMING ENVIRONMENT FOR EQUIPPING CONSTRUCTION STUDENTS WITH SENSOR DATA ANALYTICS SKILLS

SUBMITTED: March 2024
REVISED: May 2024
PUBLISHED: March 2025
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2025.010

Mohammad Khalid, Ph.D. Student
Virginia Polytechnic Institute and State University, Virginia, United States
khalidm21@vt.edu
Abiola Akanmu, Associate Professor
Virginia Polytechnic Institute and State University, Virginia, United States
abiola@vt.edu
Adedeji Afolabi, Research Associate
Virginia Polytechnic Institute and State University, Virginia, United States
adedeji@vt.edu
Homero Murzi, Associate Professor
Virginia Polytechnic Institute and State University, Virginia, United States
hmurzi@vt.edu
Ibukun Awolusi, Assistant Professor
The University of Texas at San Antonio, San Antonio, TX, United States
ibukun.awolusi@utsa.edu
Philip Agee, Assistant Professor
Virginia Polytechnic Institute and State University, Virginia, United States
pragee@vt.edu

SUMMARY*: Classification of construction resource states, using sensor data analytics, has implications for improving informed decision-making for safety and productivity. However, training on sensor data analytics in construction education faces challenges owing to the complexity of analytical processes and the large stream of raw data involved. This research presents the development and user evaluation of ActionSens, a block-based end-user programming platform, for training students from construction-related disciplines to classify resources using sensor data analytics. ActionSens was designed for construction students to perform sensor data analytics such as activity recognition in construction. ActionSens was compared to traditional tools (i.e., combining Excel and MATLAB) used for performing sensor data analytics in terms of usability, workload, visual attention, and processing time using the System Usability Scale, NASA Task Load Index, eye-tracking, and qualitative feedback. Twenty students participated, performing data analytics tasks with both approaches. ActionSens exhibited a better user experience compared to conventional platforms, through higher usability scores and lower cognitive workload. This was evident through participants' interaction behavior, showcasing optimized attentional resource allocation across key tasks. The study contributes to knowledge by illustrating how the integration of construction domain information into block-based programming environments can equip students with the necessary skills for sensor data analytics. The development of ActionSens contributes to the Learning-for-Use framework by employing graphical and interactive programming objects to foster procedural knowledge for addressing challenges in sensor data analytics. The formative evaluation provides insights into how students engage with the programming environment and assesses the impact of the environment on their cognitive load.*

KEYWORDS*: Construction, Sensor, Data Analytics, Machine Learning, Usability, End-User Programming, Construction Education.*

REFERENCE*: Mohammad Khalid, Abiola Akanmu, Adedeji Afolabi, Homero Murzi, Ibukun Awolusi & Philip Agee (2025). Design and Usability Evaluation of an End User Programming Environment for Equipping Construction Students with Sensor Data Analytics Skills. Journal of Information Technology in Construction (ITcon), Vol. 30, pg. 213-242, DOI: 10.36680/j.itcon.2025.010*

1. INTRODUCTION

In recent years, the construction industry has witnessed a surge in innovative breakthroughs with sensing technologies that have the potential to revolutionize the way sensor data is generated, analyzed, and communicated (Ellis, 2020). Sensing technologies, such as inertial measurement units and global positioning systems, can improve traditional construction techniques by enabling the collection and analysis of vital activity and resource information, opening new avenues for improved project management practices (Akhavian and Behzadan, 2015). The increasing adoption of sensing technologies has led to a significant increase in the volume of sensor data generated within the construction industry (Baduge et al., 2022). With data analytics techniques, such as machine learning (ML), construction practitioners can evaluate massive volumes of construction sensor data to detect trends, anticipate outcomes, and make data-driven decisions (Liu et al., 2022). Activity recognition, an ML technique has been studied by many researchers to classify resource states such as workers' ergonomic postures and equipment performance (Martín et al., 2013, Rashid and Louis, 2019). By employing classification techniques and utilizing data from sensors such as accelerometers and gyroscopes, it is possible to classify common activities of workers (Martín et al., 2013). Framework for activity recognition can successfully identify unique data patterns using multiple inertial measurement units (IMUs) attached to target resources highlighting the potential for operational activity recognition in construction (Rashid and Louis, 2019).

Extraction of valuable insights from the vast amount of data generated by sensors requires skills to understand and apply analytics to the data (Krishnamurthi et al., 2020). However, construction organizations face hurdles in recruiting skilled personnel with appropriate capabilities, which limits their capacity to capitalize on the complete potential of sensor data analytics for improving project outcomes (Cheng et al., 2013, Mansouri et al., 2020). Adepoju and Aigbavboa (2021) have recognized a notable deficiency in expertise related to data analytics within the construction industry. Furthermore, the authors have emphasized the necessity for training programs aimed at cultivating a proficient and knowledgeable workforce in these areas. The insufficient attention given to topics related to sensing technologies in construction education (Ogunseiju et al., 2021) and the lack of interactive platforms where students can analyze construction-related sensor data (Rowe et al., 2020) may have significantly contributed to the shortage of skilled workforce proficient in sensor data analytics (Mansouri et al., 2020). A thorough understanding of computational concepts, as well as proficiency in sensor data analytics, are critical in effectively processing, analyzing, and presenting the results of the large volume of sensor data acquired from construction sites (Akanmu et al., 2022). Navigating the complexities of sensor data analytics also necessitates proficiency in areas such as familiarity with data collection methods, data preprocessing, feature extraction, statistical analysis, machine learning algorithms, and data visualization (Ngo et al., 2020).

Educators and researchers in other fields have acknowledged the benefits of End User Programming (EUP) or End User Development (EUD) supported block-based programming environments (BBPEs) as an effective way to enhance learners' domain-specific skills and computational thinking (CT) in academic and professional settings (Rahaman et al., 2020, Zhong, 2013, Glas et al., 2023). EUD is a human-centered methodology that complements user-centered and participative design, while EUP is a sub-area of EUD that specifically concentrates on software coding. Block-based programming, on the other hand, is a technique within the realm of EUP that simplifies coding using visual blocks (Coronado et al., 2021). Block-based programming offers a user-friendly and visually intuitive interface, allowing individuals with no programming experience to easily design and modify data analysis workflows. The complications of technical syntax and code are avoided by using a drag-and-drop technique, allowing domain experts to focus on the logical and structural parts of their data analytics assignments (Bau et al., 2017). Nevertheless, the development and customization of BBPEs to cater to specific user communities introduce a potential risk: if the end-users do not find the user experience satisfactory, it may undermine the effectiveness and adoption of the BBPEs. Assessing the content and usability factors is crucial to designing an efficient learning environment that effectively serves educational purposes (Glas et al., 2022). Hence, formative assessment assumes a significant role in identifying and addressing usability-related issues within Human-Computer Interaction (HCI) platforms (Karat, 1997). By providing feedback and evaluation from end-users, formative assessment enables developers to iteratively enhance the user experience and functionality of these platforms.

To fill this gap the study designed and performed a formative evaluation of a BBPE, called ActionSens, which is specifically designed to perform sensor data analytics such as activity recognition in construction. The formative evaluation includes a comparison of the usability of ActionSens with a traditional method involving a combination of Excel and MATLAB (Ex-MAT). By conducting this evaluation, the research aims to assess the usability of the

BBPE in comparison to existing platforms for analyzing sensor data in construction education. In this study, four measurements are employed: (a) the System Usability Scale (SUS), which evaluates the overall usability of the systems, (b) the NASA Task Load Index (TLX), which measures perceived workload, and (c) eye-tracking technology, which tracks users' visual attention and information processing time, and (d) qualitative responses through semi-structured interviews. The design and usability study of BBPE in construction education has important implications since it emphasizes the necessity for user-friendly tools to improve student learning. BBPE offers an opportunity to close the gap between traditional engineering education and the skills required for sensor data analytics in construction decision-making. Integrating BBPE into construction education could help students develop their computational thinking abilities and prepare them for the field's rising technology needs. In Section 2, background information is provided on the relevant concepts. Section 3 describes the methodology employed, including the development of the environment, the experimental procedures, and data analysis. The results of the experiment are presented in Section 4. Finally, the conclusion synthesizes the findings, addresses study limitations, and discusses the practical implications of implementing the pedagogical platform in real-world learning environments, such as university classrooms and computer labs. The study contributes to knowledge by illustrating how construction domain information can be embedded in block-based programming environments to prepare students with the skills to perform sensor data analytics. The design of the BBPE, ActionSens, contributes to the Learning-for-Use framework through the use of graphical, interactive programming objects to develop procedural knowledge for addressing sensor data analytics problems. The formative evaluation illustrates how students allocate their attention within the programming environment and assesses the environment's influence on their cognitive load.

2. BACKGROUND

2.1 Construction Sensor Data Analytics

The complexity of construction projects having dynamic activities requires precise monitoring and analysis of pertinent information for efficient and successful completion. The traditional method involving manual inspections, data collection, and processing can often fall short of delivering accurate and timely information and limits the opportunities to apply analytics to make decisions accounting for productivity, safety, and quality of construction activities (Shen and Lu, 2012). Sensing technologies such as laser scanners, cameras, drones, Inertial Measurement Units (IMU), Global Positioning Systems (GPS), Ground Penetrating Radar (GPR), and Radio Frequency Identification (RFID) have emerged as potential interventions for breaking beyond these constraints (Arabshahi et al., 2021). The advancements in sensing technologies have led to the accumulation of large volumes of data, which contains valuable information that can be utilized to address various construction issues (Anumba et al., 2021). However, despite large sensor data streams, useful knowledge and information must be first extracted utilizing advanced analytics techniques such as ML, data mining, and statistical models (Aggarwal, 2013, Heureux et al., 2017). By analyzing the sensor data and identifying patterns, trends, and anomalies, construction professionals can make informed decisions regarding project planning, resource allocation, risk mitigation, and quality control (Mansouri et al., 2020). Extensive research has focused on utilizing ML techniques to recognize a wide range of construction activities. The advancements in wearable sensors and mobile devices have introduced kinematic-based approaches to identify various kinematic patterns of actions taken by construction personnel and equipment by using a variety of sensors, including accelerometers and gyroscopes. These sensors can be integrated into a microfabricated electronic chip, such as an IMU, to gather data that, when processed, could provide details about the rotational speed and orientation of workers or machinery (Sherafat et al., 2020).

Leveraging unique kinematic signals such as body acceleration, angular movement, and posture allows precise monitoring and classification of construction workers' actions, bringing substantial advantages to the construction industry in terms of safety, productivity, ergonomics, and quality control (Sherafat et al., 2020). As a result, sensor data analytics has become vital for converting the unprocessed data collected with sensing technologies into knowledge that can be used to improve construction practices. Processing and extracting meaningful insights and knowledge from large amounts of data can be a challenging task, which requires careful analysis to make informed decisions (Liu et al., 2022). However, there exists a significant gap in the provision of sufficient training to the workforce for the development of skills in construction-related data analytics approaches (Mansouri et al., 2020). Construction curricula are generally not tailored to specialize learners in sensor technologies. As a result, a substantial part of the future construction workforce could miss out on leveraging the advantages of utilizing sensor data analytics to enhance construction operations. Improving construction students' sensor data literacy requires the adoption of technologically advanced strategies that can create the opportunity to engage in analytics and develop a thorough comprehension of real-world construction sensor data sets (Rowe et al., 2020).

2.2 End-User Programming Environment

In recent times, innovative technologies have made significant inroads into the field of education, offering students enhanced and personalized learning experiences. Among these technologies, EUP concepts have gained recognition for their high customizability and value in educational settings. EUPs offer user-friendly platforms that are accessible to a broad range of learners, including those without any prior programming experience (Barricelli et al., 2019). These systems have proven to be highly beneficial for closing skill gaps across diverse academic fields, including chemistry (Zhong, 2013), biology (Gupta et al., 2017), physics (Galan et al., 2017), robotics (Rahaman et al., 2020), cybersecurity (Glas et al., 2023), and data science (Olney and Fleming, 2019). EUP environments can aid students in acquiring domain-specific knowledge while developing their computational thinking skills (Rahaman et al., 2020, Zhong, 2013, Glas et al., 2023). Block-based programming environments (BBPEs) are a notable concept within EUP, and their growing recognition can be attributed to their visually driven programming interface which makes programming more intuitive and accessible, especially for users without prior coding experience. BBPEs leverage interactive blocks to symbolize codes and programming concepts, providing users with versatile drag-and-drop capability. These features make BBPEs highly user-friendly, allowing individuals to easily grasp and implement complex computational workflows. BBPEs allow users to emphasize the logical structure and functionality of their algorithms based on semantics rather than syntax or other programming language intricacies (Bau et al., 2017).

The integration of Computational Thinking (CT) skills through the use of BBPE environments has shown promising results and is supported by a substantial body of research. For instance, Gupta et al. (2017) implemented a BBPE called BioBlocks to address the challenge of reproducibility in academic biology experiments. By utilizing BioBlocks, the authors aimed to reduce ambiguity and minimize human error in experimental protocols. Sarmento et al. (2015) conducted a study wherein students from diverse academic disciplines such as chemistry, mechanical engineering, and electrical engineering were involved in using BBPE as a platform for developing CT skills and solving problems related to sensors and robots. The results demonstrated a positive impact on various motivational aspects, such as increased attention, improved relevance, and enhanced confidence levels among the participants. In a study by Tawfik et al. (2022), the use of a BBPE was investigated as a medium to educate adult learners on data science skills. The study's conclusions indicated that the blocks within the programming environment not only served as useful visual aids but also significantly contributed to the learners' ability to comprehend CT principles. Despite these studies, there is a dearth of research on the usability and effectiveness of block-based learning tools, highlighting a significant gap in the design and development of tools that facilitate CT at various educational levels (Rijo-García et al., 2022). Due to the scarcity of evaluations of usability, it can be difficult to completely comprehend user reactions to these highly specialized applications incorporating ML techniques (Chen et al., 2021).

2.3 Evaluation of User Experience

Considering the manifold complexity of interactive systems, evaluation of a newly built computer interface for a specific user population is essential to ensuring a smooth user experience. Formative assessment plays a vital role in ensuring the usability of a system by involving users and collecting their input during the development and design phases. By obtaining early input, developers can identify specific areas that require improvement, thereby minimizing the need for significant revisions in the final stages. This iterative approach allows for continuous enhancement of the user experience throughout the development process (Rosson and Carroll, 2002). The following describes measures adopted for evaluating the proposed EUP environment.

2.3.1 Overall System Usability Score

The System Usability Scale (SUS) is a popular subjective measurement approach that utilizes a standardized questionnaire to assess the usability of a range of systems by gathering user perceptions and feedback. The SUS's impartiality enables it to evaluate a wide range of user interfaces, such as websites, mobile devices, interactive platforms, TV applications, and more (Derisma, 2020). The SUS offers features, including a condensed ten-question style that makes it quick and simple for participants and administrators to complete and score. Assessing a user interface's ease of use during design is essential, and poor usability often results in user abandonment of interactive systems (Derisma, 2020). In pedagogical contexts, usability factors may be of even greater importance. Derisma (2020) conducted a usability assessment of the CodeSaya.com Portal, an Information and Communication Technology or ICT-based medium intended to facilitate teaching and learning activities. Utilizing the SUS, they obtained a benchmark score that serves as critical recommendations for the future development of online learning platforms. Dawoud et al. (2021) employed the SUS to evaluate and compare the usability of collaborative and individual visual programming on a block-based programming platform. Their findings indicated that collaborative programming exhibited superior usability performance compared to participants who engaged in solo coding.

2.3.2 Perceived workload

Cognitive load, within the realm of HCI, refers to the extent of mental exertion or resources required to carry out a task during computer system interactions. In practical terms, any cognitive task relies on an individual's working memory. The Cognitive Load Theory (CLT) underlines the consideration of working memory limitations in instructional design to avoid deterioration in learning performance (Paas and Sweller, 2014). The presentation of computational concepts should prioritize minimizing cognitive load while maximizing their pedagogical significance (Tudoreanu, 2003). The NASA Task Load Index (NASA TLX) proves to be highly useful in assessing the cognitive load experienced by users as it provides a multidimensional approach to subjective workload measurement (Hart and Staveland, 1988). The measurement tool takes into account the user’s experience-related factors such as mental, physical, and temporal demands, as well as effort, frustration, and performance levels. The results can aid researchers and designers in comprehending the impact of a task on users' cognitive and physical demands, as well as its effects on the overall user experience. Assessing the cognitive load of specific purpose-built HCI systems across diverse domains is of utmost importance before its deployment, given the increasing complexity of computer interfaces and interactions that expose end-users to mentally challenging tasks (Kumar and Kumar, 2016). Prior studies in the domain of block-based EUP have used NASA-TLX for the evaluation of cognitive load. For instance, using NASA-TLX, Dawoud et al. (2021) indicated that the collaborative programming setting led to a decrease in the users' overall cognitive load compared to the individual programming setting. In a cyber security training study, post-training feedback using NASA TLX was utilized to identify the specific stages where the visual programming language (VPL) positively influenced the learning experience for participants (Glas et al., 2022). The VPL platform employed a block-based interface, allowing for the measurement of trainees' perceived workload throughout the entire learning process. The experimental group utilized the VPL (i.e., Blockly), as compared to the control group using the text-based language (JSON). Interestingly, both groups reported equally positive learning experiences, although participants in the VPL group found the learning process more enjoyable indicating the potential of VPL usage for other domains. Pratidhina et al. (2021) investigated the potential advantages of visual programming (i.e., block-based environment) by comparing it to a conventional text-based language. The outcomes of the NASA-TLX scores suggested that visual programming environments provided a lower perceived workload, a more favorable user experience, and more perceived success among adult end-user programmers.

2.3.3 Visual attentional resources

Eye-tracking technology has demonstrated its effectiveness in evaluating the usability of computer interfaces. Eye tracking refers to the method of recording the positions and movements of the eyes in relation to visual stimuli (Bojko, 2005). Eye-tracking technology has become more affordable and accessible, enabling researchers to examine eye movements and collect informative metrics with improved precision and effectiveness (Pernice and Nielsen, 2009). Therefore, eye tracking has been widely accepted within enlarged research communities for usability evaluation purposes (Goldberg et al., 2002). By employing fixation metrics within specific areas of interest (AOIs), researchers can monitor and examine users' eye movements while they engage with a computer interface (Brunyé et al., 2019). This enables researchers to gain insights into participants' visual attention patterns and delve into their cognitive processes and decision-making strategies during the interaction (Barral et al., 2020). As a result, it allows for the identification of behavioral and interaction patterns exhibited by end-users (Lai et al., 2013). Multiple research studies have provided evidence that high cognitive load is related to high fixation durations (Park et al., 2015, Korbach et al., 2016) and high fixation counts (Van Orden et al., 2001, Van Orden et al., 1998). However, it should be noted that depending on the context, higher fixation durations can also be due to deeper cognitive processing or interest in the visual stimuli (Poole and Ball, 2006, Lee et al., 2019) (see **Error! Reference source not found.**). Eye-tracking has been used as a measure of usability in various studies related to HCI, but there is a lack of research on its application in assessing the usability of block-based environments.

2.4 Theoretical Framework

The development of ActionSens, a block-based programming environment (BBPE), draws its theoretical underpinning from the Learning for Use (LfU) theory, which serves as the basis for technology-driven platforms aimed at fostering learners' skill development and facilitating meaningful comprehension (Edelson, 2001a). This theory is based on four the following four tenets: “(1) knowledge construction is incremental; (2) learning is goal-directed; (3) knowledge is situated; and (4) procedural knowledge needs to support knowledge construction” (Edelson, 2001b). These tenets encourage designers to create units centered around knowledge application tasks that generate a need for learning objectives. They also allow learners to enhance their understanding by applying their newly acquired knowledge and skills, thus refining their abilities. The tenets are applied to the development of ActionSens by ensuring a goal-directed problem-solving approach and progressive knowledge acquisition. ActionSens was structured around a hierarchical workflow of standard data analytics procedures. These procedures or tasks were carefully arranged as usability benchmarks to provide researchers with a basis for comparing task performance across different situations. The benchmark tasks include i) data selection, ii) data transposing, iii) data merging, iv) data labeling, v) data splitting, vi) data pre-processing, and vii) ML training. This hierarchical arrangement allows learners to follow a structured path, starting with simpler tasks such as data selection and gradually progressing towards more complex tasks (involving further processing of the data sets), such as data merging, labeling, splitting, pre-processing, and ultimately ML training. While learners actively participate in exploring and reviewing construction activities and the corresponding sensor data that captures pertinent activity information, they concurrently develop and expand their analytics abilities using block representations. This process involves leveraging previously acquired data structures from earlier stages which aligns with the first and fourth tenets of LfU, which emphasize that the acquisition of new knowledge is a gradual and incremental process. Moreover, the structured workflow for ML classification, utilizing construction sensor data to derive actionable insights such as prediction results and performance metrics, acts as a sequential roadmap for performing the data analytics tasks. This approach aligns with the second and third tenets of LfU theory, emphasizing that knowledge acquisition is goal-oriented and context-dependent. The analytics workflow in the platform supports a systematic and logical approach to solving goal-directed tasks, allowing learners to build upon their acquired knowledge and skills in a step-by-step manner. The hierarchical structure facilitates a logical progression, guiding users to advance from basic data manipulation to more advanced analytical techniques, all with a strong focus on utilizing these insights to enhance decision-making in construction projects. Lastly, the LfU theory incorporates cognitive theories of learning, including Cognitive Load Theory (CLT), which emphasizes managing cognitive load for optimal learning outcomes. CLT posits that instructional design should consider the constraints of working memory to avoid overwhelming its capacity and hindering learning (Sweller, 1988).

2.5 Research Gap

While block-based learning environments have gained widespread acceptance in other educational domains for their effectiveness in developing targeted skills, construction education lags in their adoption. Specifically, there is a significant gap in using block-based learning environments to train students and the workforce on sensor data analytics. The potential benefits of incorporating EUP as a pedagogical platform in construction education such as sensor data analytics have not been extensively explored, leading to a limited understanding of the interaction factors that can influence user experience outcomes. This limitation also hampers the further improvement of such environments. With no benchmarks for controlling or guiding the cognitive processes of the end-users, lack of user experience factors identification leaves the users at the potential of delinking the system leading to rejection.

3. METHODOLOGY

This section describes the approach employed to design and develop ActionSens, the experimental details, the participants involved, and the methods used for data collection and analysis (Figure 1). The evaluation compares the usability of ActionSens with a combination of traditional platforms, Microsoft Excel and MATLAB (Ex-MAT), typically used to perform similar data analytics tasks. In simpler terms, the purpose was to determine whether ActionSens improves the users' analytics process by making it more efficient in terms of requiring less mental effort, managing a better allocation of attentional resources, and ultimately providing an improved user experience.

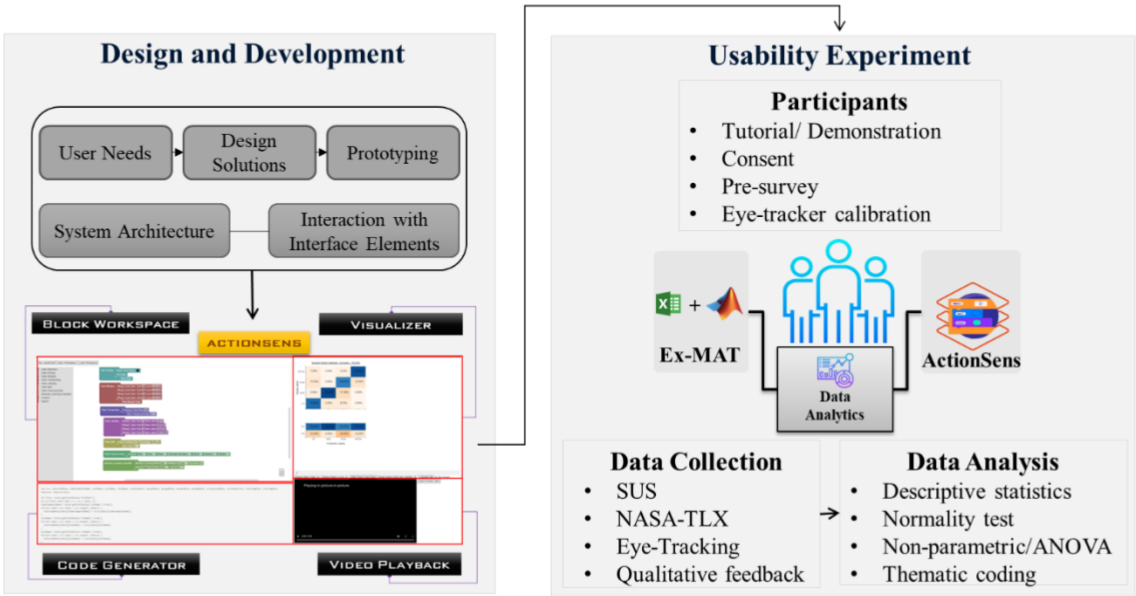


Figure 1: Overview of the research methodology (image/icon source: Freepik).

3.1 Development of ActionSens

This section describes the design and development process of ActionSens, which adopted the agile User Experience (UX) lifecycle methodologies (Hartson and Pyla, 2012). The agile UX lifecycle assessment strategy places a strong emphasis on user input, iterative design, and collaboration between designers, developers, and end users. The lifecycle assessment involves user research, design solutions, prototyping, and evaluation as key stages in the development process.

3.1.1 User research

In a previous study (Khalid et al., 2023), an extensive industry survey was conducted to acquire an understanding of the expectations of end-users and the industry's prerequisites concerning the utilization of sensor data analytics in construction education. This survey was validated by a focus group of construction industry professionals. Understanding user needs helps define specific features for the system that align with user-centered design concepts (Hartson and Pyla, 2012).

3.1.2 Creation of design concepts

Focusing on the results of the user research phase, an ideation and creation phase was initiated to establish the objectives and specifications for the block-based ActionSens interface. This phase involved brainstorming to develop ideas, sketch, critique, and finally synthesize the outcomes as early wireframes (Hartson and Pyla, 2012). This further involved creating user personas of construction students and composing user classes and roles, workflow modeling, and tasks. As a part of design concept creation, numerous wireframes were developed to model the workflow and corresponding operations required for the data analytics task performance. Block-based environments should be constructed based on design frameworks to enable a better user experience (Karakasis and Xinogalos, 2020). Therefore, an End-User Development (EUD) design framework was adopted as a guideline for the design that also aligns with the requirements of the data analytics tasks. Barricelli et al. (2023) presented an EUD design framework, indicating their features to enhance end-users' CT skills within the platform which allowed the researchers to incorporate the basic features that characterize a web-based block-based environment. This framework was chosen as it not only supports the enhancement of students' CT skills but also facilitates the execution of sensor data analytics tasks within the platform.

3.1.3 Prototyping

Low-fidelity prototypes were developed using tools such as digital sketches and Blockly customized blocks. Initially, customized blocks were programmed to perform specific operations within the ML workflow, such as data selection, merging, transposing, and labeling. These customized blocks were rigorously tested to detect any potential interaction design issues, including the actions users would perform and the information they would view to effectively advance to the progressive levels of the ML workflow (**Error! Reference source not found.**).

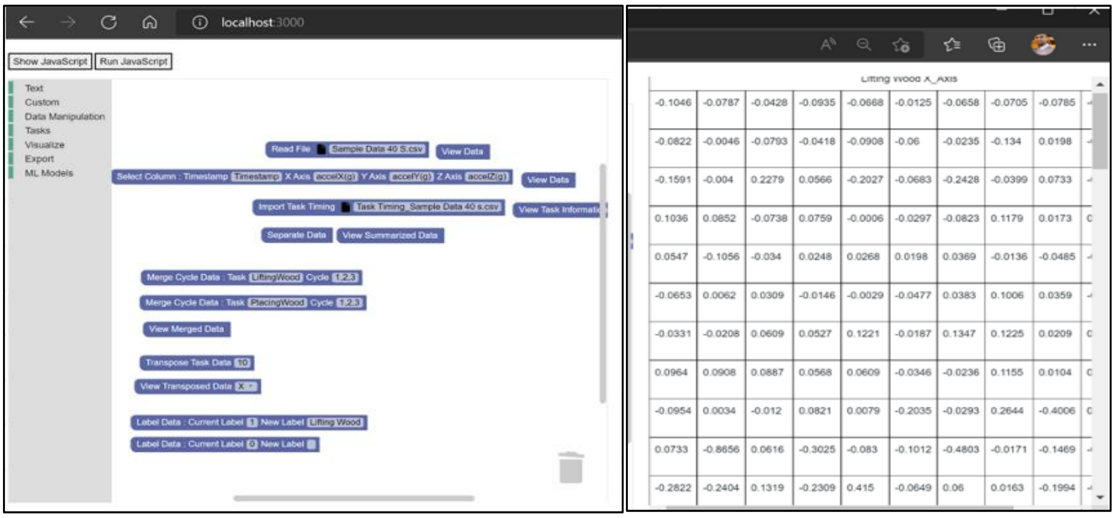


Figure 2: Testing with customized blocks (left) and data visualizer (right) to identify interaction design issues using a high-fidelity prototype.

These prototypes were shared with three researchers for feedback which allowed for early identification of potential issues and opportunities for improvement before advancing towards a more detailed design of a high-fidelity prototype. For prototyping, this research adopted a “T” prototype that combines the advantages of both the horizontal and vertical prototypes. The horizontal prototype was effective in demonstrating the product concept tested which contained broad features it incorporated but offered less depth in its coverage of how that functionality works (Kensing and Munk-Madsen, 1993). The vertical prototype offered a comprehensive level of detail for a specific set of features, enabling a thorough understanding of individual interaction workflows and their practical implementation. This depth of functionality proves beneficial when representing and comprehending isolated parts of the workflow, ensuring a complete grasp of how these details are utilized in real-world scenarios (Hartson and Pyla, 2012).

3.2 Evaluation

3.2.1 Participants

To conduct the usability experiment, a group of twenty (20) undergraduate students (i.e., 11 males and 9 females) was selected through recruitment methods such as the university's listserv and flyer distribution. Previous studies emphasize that testing with five participants is enough to detect 80% of system usability issues (Virzi, 1992, Lewis, 1994, Rough, 2018). Similar studies have used less than 20 participants in their usability studies (Lucas and Thabet, 2008; Irizarry et al., 2012). To be eligible for participation in this study, the individuals had to meet specific inclusion criteria, which included being undergraduate students pursuing majors in civil engineering, building construction, or construction engineering management, and being at least 18 years of age.

3.2.2 Data collection

Participants were provided with surveys to collect demographic information, subjective data, and feedback. Before the evaluation, participants completed a pre-survey to gather data on age, gender, educational background, and experience with similar platforms. After completing each round of tasks (without and with ActionSens), participants were asked to complete the SUS questionnaire to assess their perception of usability. To measure perceived workload during task performance, the NASA-TLX questionnaire was administered. Eye tracking data, including fixation information, were collected to analyze participants' eye movements during their interactions with the platforms. Participants' eye movements were recorded using an eye tracker (Tobii Pro Glasses 3). Upon completion of each task performance, qualitative data were collected via semi-structured interviews.

3.2.3 Experimental procedures

The experiment employed a two-task performance approach. Participants were assigned to two different conditions for completing data analytics tasks: one condition involved using a combination of Microsoft Excel and MATLAB (Ex-MAT), while the other condition involved using ActionSens. This approach ensured that participants could reach comparable conclusions through both conditions. The experiment utilized a repeated measure within-participant design to develop the comparisons. A break of approximately 20-30 minutes was taken by the participants before transitioning to the second task, allowing them to rest and refresh before engaging in the subsequent task. Before arriving for the experiment, all participants were provided with the accessible version of tutorial materials that provided comprehensive information about the task workflows, along with essential components and notable features (Ramoğlu et al., 2017). Participants received a 15-minute practical demonstration upon arrival on the basic workflows of the platforms, the construction activity video, and the raw sensor dataset. This demonstration served to familiarize participants with the step-by-step procedures involved in the tasks. Following the approved Institutional Review Board (IRB) protocol, at first, the informed consent form was presented, and the pre-survey responses were recorded.

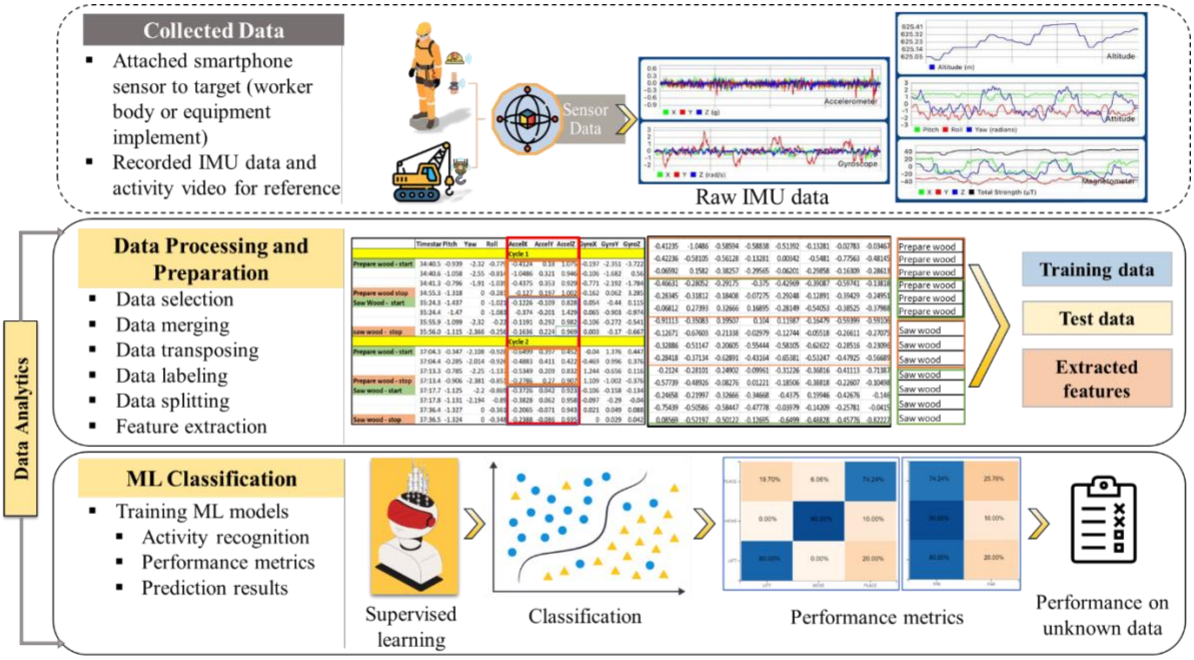


Figure 3: Overview of data analytics workflow (image/icon source: Freepik).

The experimental setup involved configuring computer systems to run the platforms under evaluation. To ensure uniformity across participants, the hardware requirements and software versions were tested for uninterrupted operations before experiments. Participants used a highly configured desktop computer for the task performance, while additional software and hardware for data capture (such as eye tracking) were carried out on a separate laptop to monitor the recording of the data. The participants were seated in a controlled setting to ensure comfort, maintain consistency throughout the evaluation, and eliminate any potential discomfort or distractions. Participants received a briefing on the procedures for collecting eye-tracking data with Tobii Pro Glasses 3. Before providing the participants with the eye-tracking equipment, the trackers were cleaned and adjusted using different nasal bridges to verify that each participant had a suitable fit. As any alterations throughout the experiment might affect calibration, participants were advised to confirm the comfort and fit of the glasses. Before the evaluation, calibration procedures were conducted to ensure accurate eye-tracking measurements and the recording was only initiated once acceptable calibration was achieved.

The tasks assigned to participants involved interacting with pre-recorded construction activity information, including activity video recordings and corresponding raw IMU sensor data (see Figure 3). The raw IMU data, which represented actual movement data, was captured previously using a mobile application called SensorPlay during a mimicked construction activity involving lifting and placing materials. Participants were tasked with processing the raw sensor data and training ML models to achieve classification capabilities and evaluate prediction performance such as confusion matrix and comparisons between predicted and actual data.

Table 1 presents a comprehensive overview of the tasks completed by participants on both platforms.

Table 1: Data analytics tasks completed by participants.

Task Workflow	Ex-MAT	ActionSens
Data selection	• Excel: Retain required data columns in a spreadsheet • Excel: Sort data in a spreadsheet based on pre-specified construction activity information (i.e., different tasks, timestamps, and cycles)	• Read File: Read raw IMU data from the local drive • Data Selection: Retain required data columns • Data Sorting: Sort data based on pre-specified construction activity information (i.e., different tasks, timestamps, and cycles)
Data merging	• Excel: Merge data into corresponding tasks and create individual spreadsheet files for each task	• Merge multiple cycles of data under the same task
Data transposing	• MATLAB: Apply a predefined code to transpose data into a specified number of columns and generate separate spreadsheet files as output	• Transpose of data into a specified number of columns.
Data labeling	• Excel: Assign appropriate labels to correspond with specific tasks within the data in the spreadsheet file	• Specify data labels for the tasks.
Data splitting	• Excel: Split the data into separate sets for training and testing purposes, and proceed to copy each set into individual spreadsheet files	• Segment the data into training and testing portions, allowing for customization based on user-defined input.
Data pre-processing or feature extraction	• MATLAB: Import all the training spreadsheet files and implement a predefined code that generates a table of statistical features from the data	• Select the statistical features to be extracted from the data.
Machine learning classifiers or model training and testing (ML training)	• MATLAB: Train ML models and generate confusion matrix; Export trained model; Import testing spreadsheet files; Modify code to generate a table of prediction results	• Select the models to be trained by specifying the validation schemes, such as cross-validation or holdout; Evaluate the models by examining the confusion matrix; Test the trained model to observe the prediction results.

3.3 Data Analysis

The data obtained from the SUS and NASA TLX questionnaires for both conditions (i.e., Ex-Mat and ActionSens) were treated as ordinal, while the eye-tracking data was considered continuous. To assess the data distribution, the Shapiro-Wilk test was performed, indicating that most data did not follow a normal distribution. To account for the violation of the normality assumption, Wilcoxon Signed-Ranks Tests (WSRT) were employed to determine the presence of statistically significant differences between the dependent variables, including SUS, NASA, and eye-tracking metrics. The independent variables considered were the Ex-MAT and ActionSens conditions. A p-value of less than 0.05 was considered to be significant. Descriptive statistics, including mean, median, and standard deviations, were computed to present the comparisons in a visual format and summarize the results.

3.3.1 System Usability Scale (SUS)

The scoring process for the SUS questionnaire is as follows: Odd-numbered questions are scored by subtracting 1 from the user score, while even-numbered questions are scored by subtracting the user score from 5. The final SUS score for each participant is obtained by multiplying the sum of these scores by 2.5 (Sauro, 2011). To calculate the mean SUS score for multiple participants, the total SUS scores of each participant are added together and then divided by the number of participants (Derisma, 2020).

3.3.2 NASA-TLX

The data obtained from the NASA-TLX questionnaire included subscales such as mental demand, physical demand, temporal demand, performance, effort, and frustration (Hart and Staveland, 1988). In addition to mental, frustration, and effort, the importance of considering physical, temporal, and performance demands in visual programming tasks is highlighted by Almusaly et al. (2018). For instance, in tasks involving drag-and-drop operations, users must maneuver blocks to place them accurately, adding to both physical (i.e., number of mouse or keyboard clicks) and cognitive workload. Furthermore, the time required to compose a program in a block-based interface, influenced by the number of blocks involved, contrasts with the efficiency of text-based alternatives, illustrating the importance of evaluating temporal demand in such contexts. Moreover, the abundance of options within blocks, altered through small icons, can slow down the entry process, potentially frustrating users and impacting performance.

Each workload sub-scale is divided into 20 equal intervals, denoted as "low" and "high" at both ends. The NASA (raw) Task Load Index, or RTLX, simplifies workload assessment by omitting pairwise comparisons from the original TLX, facilitating a direct average calculation across six dimensions and exhibiting robust experimental validity (Georgsson et al., 2019). To standardize results on a 0–100 scale, the score calculation formula is (rating-1) and multiplied by 5. The participant's overall cognitive workload, represented by the RTLX score, is derived by summing the total scores across the six dimensions and dividing them accordingly. This fundamental approach of RTLX indicates that higher summed averages correspond to elevated experienced cognitive workload, as described by Lovasz-Bukvova et al. (2021). The scores from all the participants were averaged and the mean score of each sub-scale and RTLX score were reported.

Table 2: Relevant eye-tracking metrics.

Eye-tracking Metrics	**Cognitive process or usability-related issues**	
Total fixation duration in AOI	Longer fixations relate to difficulty in extracting information, or it means the media is more engaging	(Wang et al., 2014, Goldberg and Kotval, 1999, Pachman et al., 2016, Pan et al., 2004)
Total fixation count in AOI	Higher fixation count relates to less efficiency in search (perhaps due to sub-optimal interface layout)	(Wang et al., 2014, Goldberg and Kotval, 1999, Pachman et al., 2016, Pan et al., 2004)
Total visit duration (dwell time) in AOI	Longer visit durations indicate difficulty in extracting information or possible importance of the element	**(Jacob and Karn, 2003, Borys and Plechawska-Wójcik, 2017a)**
Total visit count (dwell count) in AOI	Higher visit counts relate to confusion or possible importance of the element	**(Jacob and Karn, 2003, Borys and Plechawska-Wójcik, 2017a)**

3.3.3 Eye-tracking

Tobii Pro Lab Dynamic AOI (Area of Interest) and metrics tools were used to obtain the desired eye-tracking metrics. Seven comparable key steps were established as benchmark tasks (see **Error! Reference source not found.**) within both situations of participants, and these served as the comparable basis for mapping the AOIs on different steps. Specific fixation-related metrics were captured for each specific task step by activating the AOIs at specific timeframes. The entire recordings were carefully reviewed and then thoroughly evaluated by researchers. In cases where the AOIs went out of range due to head movement, the dynamic AOI feature was employed to ensure a more precise and accurate recording of the gaze data from the screen. One participant's eye-tracking data was excluded due to data invalidity and inaccessibility, resulting in a total of 19 participants included in the eye-tracking analysis. Furthermore, a sample size of 20 is thought to be reliable for quantitative eye-tracking research (i.e., fixation metrics), removing many of the erroneous findings and offering a narrow confidence interval (Pernice and Nielsen, 2009). A set of metrics, shown in Table 2, were extracted from Tobii ProLab. The metrics help to understand participants' visual attention to specific AOIs. These can also be used to infer the cognitive load and usability of the EUP platform (Ehmke and Wilson, 2007, Borys and Plechawska-Wójcik, 2017b). Tobii Pro Lab offers eye-tracking metrics based on pre-processed data generated by the I-VT gaze filter. These metrics can be directly exported from the software as they are already calculated. For instance, fixation count for a participant represents the number of fixations occurring within a specified time interval and within a target AOI while, while fixation duration denotes the elapsed time, measured in seconds, between the initial and final gaze points in a sequence of gaze points forming a fixation. Similarly, visit duration, measured in seconds, represents the elapsed time between the onset of the first fixation on the AOI and the offset of the last fixation and is provided for each participant. Similarly, the software provides numeric counts of the number of visits occurring within a time interval, specific to a target AOI, for each participant.

3.3.4 Verbal feedback

Following the review of the interview transcript, a de-identification procedure was implemented to protect the participants' personal information by assigning random numbers to everyone. This step ensured the exclusion of any sensitive or identifiable data. The NVIVO v.14 software was employed for qualitative data analysis, where suitable codes were assigned to the transcript. The process of open coding was utilized to identify emerging themes, based on pertinent comments extracted from the participants' responses. This approach adhered to the prescribed methodology mentioned in the guidelines by Saldaña (2009). The generated themes were used to cluster coded responses that aligned (Hsieh and Shannon, 2005). To classify and condense the data, common themes were discovered in all the interviews. The extracted themes were cross-referenced with the original transcripts to maintain consistency. To ensure the credibility of the findings, the researchers discussed and reached a consensus on the interpretation of codes and emerging themes (Miles et al., 2018, Robson and McCartan, 2016). Two researchers independently assessed the assigned codes, themes, and corresponding excerpts which resulted in an inter-rater agreement of acceptable Cohen-Kappa scores of 0.64 and 1.0 respectively for Ex-MAT and ActionSens.

4. RESULTS

In this section, the results of the design and development of ActionSens, along with the evaluation which includes the participant demographics and a comparison between ActionSens and Ex-Mat, are presented. The following aspects were compared: (a) system usability score, (b) cognitive workload, (c) eye-tracking fixation-related metrics, and (d) thematic analysis of verbal responses.

4.1 Developed Interface

4.1.1 Overview of the ActionSens platform

ActionSens was built using the Model-View-Controller (MVC), an architectural pattern used for designing web-based applications (see Figure 4). MVC uses three main layers: model, view, and controller. The roles of the layers in the design of ActionSens are described as follows:

4.1.2 Model

The model includes applications, rules, logic, and operations performed on data imported into ActionSens via the graphical user interface (GUI) and interpreted by the controller (see Section 4.2.1.3). Specifically, the model sorts, stores, and structures the data, performs ML classification processes such as feature extraction and classification and computes the performance measures. The model uses TensorFlow.js, a library for building and executing ML algorithms in web applications, for the classification of tasks and actions. The performance of the classifiers is interpreted as confusion matrices. The confusion matrices are transferred to the GUI by the view (see Section 4.2.1.2). ActionSens allows end users to choose from 7 statistical features (mean, median, mode, min, max, SD, variance), 4 ML models (logistic regression, linear regression, K-nearest neighbor, and support vector machine), and 4 performance metrics (confusion matrix, precision, recall, accuracy) that are known to be effective in construction activity recognition (Gonsalves et al., 2022). In addition, the model defines the structure and behavior of the data and exposes functions that the controller can use to retrieve and manipulate data stored in the database. The data are stored using MariaDB Server, a relational database management system. The model interacts with the database using Sequelize API (Application Programming Interface). Sequelize is a cross-platform JavaScript runtime environment mapper for facilitating interaction with databases such as MariaDB, MySQL, and SQLite.

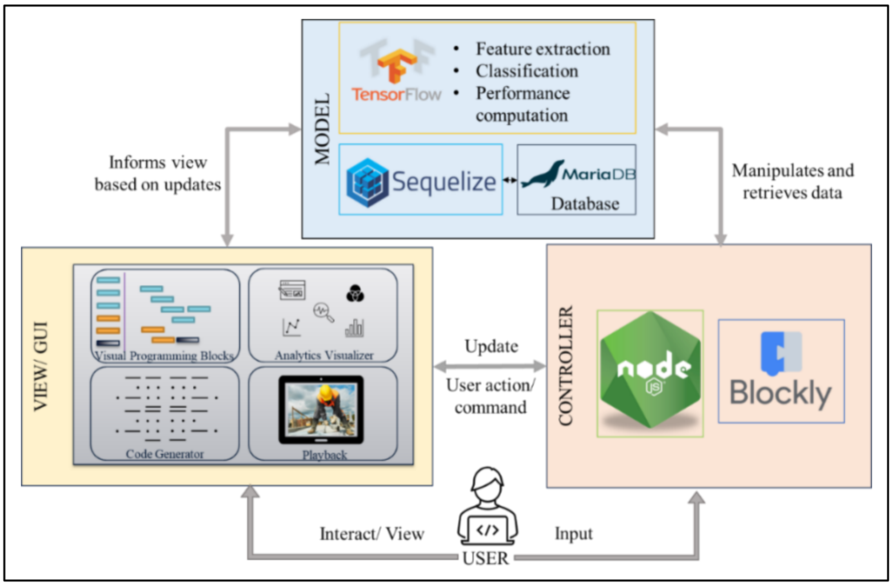


Figure 4: System architecture of ActionSens (image/icon source: Freepik).

View

The view renders information from the model onto the GUI of ActionSens. The view is to present information to learners. In addition to presenting data, the view also manages learner inputs and actions on the GUI including uploads of sensor data and task timing information, clicks on blocks, and relocation of blocks via the GUI. The view records and transmits these to the controller for processing. The view also presents results, such as structured data, videos, and confusion matrices, to learners. The view consists of the block menu, block workspace, code generator, analytics visualizer, and video playback (see Figure 5). The block menu, block workspace, code generator, and analytics visualizer were designed using Cascading Style Sheets (CSS), a style sheet language used for presenting menus on web interfaces. The video playback was embedded in the view using a Hypertext Markup Language (HTML) Video tag. JavaScript was leveraged to facilitate the capture and display of the time of the videos.

Controller

The controller updates the model and/or view in response to input from the learners. Specifically, the controller responds to learners' requests (e.g., selecting blocks from the block menu, relocating the blocks to the workspace, and executing and recycling of blocks) presented by the view in the GUI. The controller receives, validates, and transfers the requests to the model for processing. The controller of ActionSens consists of Node.js libraries such as Blockly. Node.js libraries are storages of JavaScript applications for performing specific functions. In ActionSens, Blockly libraries consist of blocks for performing different coding functions.

4.1.3 Interaction with ActionSens and connections with CT skills

ActionSens was designed to support data analytics, while also fostering learner's CT skills (Barricelli et al., 2023). Therefore, in pursuit of activating the five CT skills applied in each stage of an EUD problem, five dimensions for the platform were identified: concreteness, modularity, structuredness, reusability, and testability. This design framework also supports the general construction of Google's EUP platform Blockly (Google Inc., 2020). This section describes the key features for interacting with ActionSens and how the CT skills are relevant to the dimensions of the platform.

Block selection

This feature lets the user explore the 'Block Menu' containing a variety of blocks and select the most appropriate block for action. For example, the user can drag and drop 'Read File' on the block workspace to import appropriate raw sensor data into the interface. The users can clean the raw dataset by ‘Data Selection’ block to retain the relevant data needed for the analytics and discard the unnecessary data. As the progression takes place, the user can select the appropriate blocks for each instance from the menu as the flow of data analytics task requires (Figure 5). This feature is associated with the abstraction CT skill which is the cognitive process of selecting the most essential information about a system or situation while setting aside or simplifying the less crucial information (Calderon et al., 2022). One of the steps of problem-solving is to guide students to abstract the problem into a quantitative mathematical problem, which can be calculated by computers. The capacity of EUD environments to deliver domain-specific concepts tangibly (e.g., concrete events, conditions) without demanding highly advanced abstraction skills from the end-user is referred to as the concreteness dimension (Berti et al., 2006). The concreteness of information is a critical dimension at the initial stage for the users to view the information, confidently select the required blocks, and have them perform actions as intended.

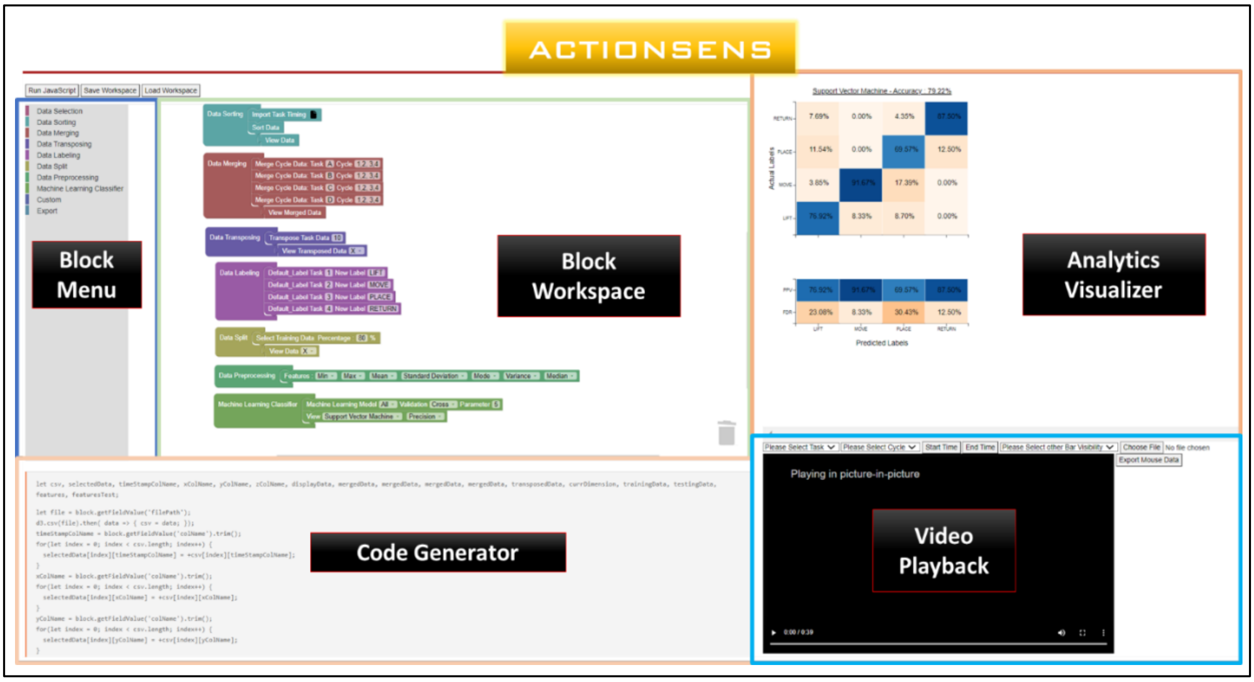


Figure 5: ActionSens interface.

Block Construction

The feature of constructing multiple interlocking blocks (also known as the container blocks that accommodate unit function blocks) in the ‘Block Workspace’ lets the user break down the entire problem into a set of manageable sub-problems through the CT’s decomposition skill (D’Alba and Huett, 2017). The availability of various elements, blocks, or modules that support end-users in decomposing a problem and identifying the pieces that may comprise its solution which can be referred to as the modularity dimension of the environment (Barricelli et al., 2023).

Block structuring

The block structuring feature enables the user to define action sequences of the analytics workflow by organizing logical connections between the building blocks to produce solutions to computational problems. This takes place within the block workspace. A general sequence of the block structuring may comprise read data, manipulate data, analyze data, and view data, for instance. This feature relates to the algorithmic thinking of CT skill, which is the method of designing and implementing algorithms to solve problems or carry out tasks (Shute et al., 2017). The structuredness dimension of EUD can be highlighted here as it refers to the environment’s ability to structure a solution in a step-by-step manner, which also simplifies the process of connecting the input and output of various steps (Barricelli et al., 2023).

Analytics results

The feature of viewing data and analytics results on the interface can be linked to the evaluation skill of CT and the testability dimension of EUD – this occurs within the 'Analytics Visualizer'. The testability dimension refers to the capacity to evaluate the results of activity within the EUD environment determining whether a solution is accurate and comparing it to other solutions to maximize it considering the available resources (Barricelli et al., 2023). For example, the environment presents a workspace screen (i.e., Analytics Visualizer) to provide visual feedback on the user's work, and the user can analyze the results and view them in a separate panel and scroll through the entire result data set to compare with the problem formulation and solving strategies. Additionally, findings may be simulated and visualized in the form of a confusion matrix and additional performance metrics such as recall and precision to evaluate the performance of the trained models.

Export results

This feature allows the user to export results in various formats as deemed appropriate for the intended application for enhanced communication with other stakeholders involved. Reusability refers to the capacity to allow the results of EUD activities to be utilized in other contexts and shared with other end-users (Barricelli et al., 2023). This relates to the generalization skills of CT which is identifying patterns in the solution of existing issues and applying the same (potentially modified) method to different problems in the future (Shute et al., 2017).

4.2 Evaluation

4.2.1 Participants demographics

The demographics of the participants are shown in Table 3. In Table 3, the result showed that there were 11 (55%) males and 9 (45%) females. A breakdown of the participants' academic program showed that most of the participants were in the Construction Engineering and Management program.

Table 3: Participants' demographic information.

Demographics	**Group** (N=20)
Gender	
• Male	55%
• Female	45%
Academic Program	
• Building Construction	20%
• Civil Engineering	35%
• Construction Engineering and Management	45%

4.2.2 Usability

The research utilized SUS questionnaires to collect subjective data, comparing the usability of ActionSens with Ex-MAT for processing sensor data. Participants rated 10 questions on a 5-point Likert scale, with odd-numbered questions reflecting positive aspects like function integration, ease of use, quick learning, and confidence in system use, while even-numbered questions addressed perceived difficulties, self-sufficiency in technical support, consistency, and learning curve. In assessing both block-based programming and traditional analytical interfaces, these elements collectively offer evidence of user preferences, efficiency, and the comprehensive usability of the interfaces examined in this study (Derisma, 2020, Dawoud et al., 2021). ActionSens received a SUS score of 86 which falls within the highest category, Grade A, as compared to Ex-MAT which obtained a usability score of 49.75 or Grade F (where A > 80.3; B = 68-80.3; C = 68; D = 51-68; F < 51) (Sauro, 2011). All measures for each subscale of SUS were compared between Ex-MAT and ActionSens conditions using mean scores as presented in Figure 6.

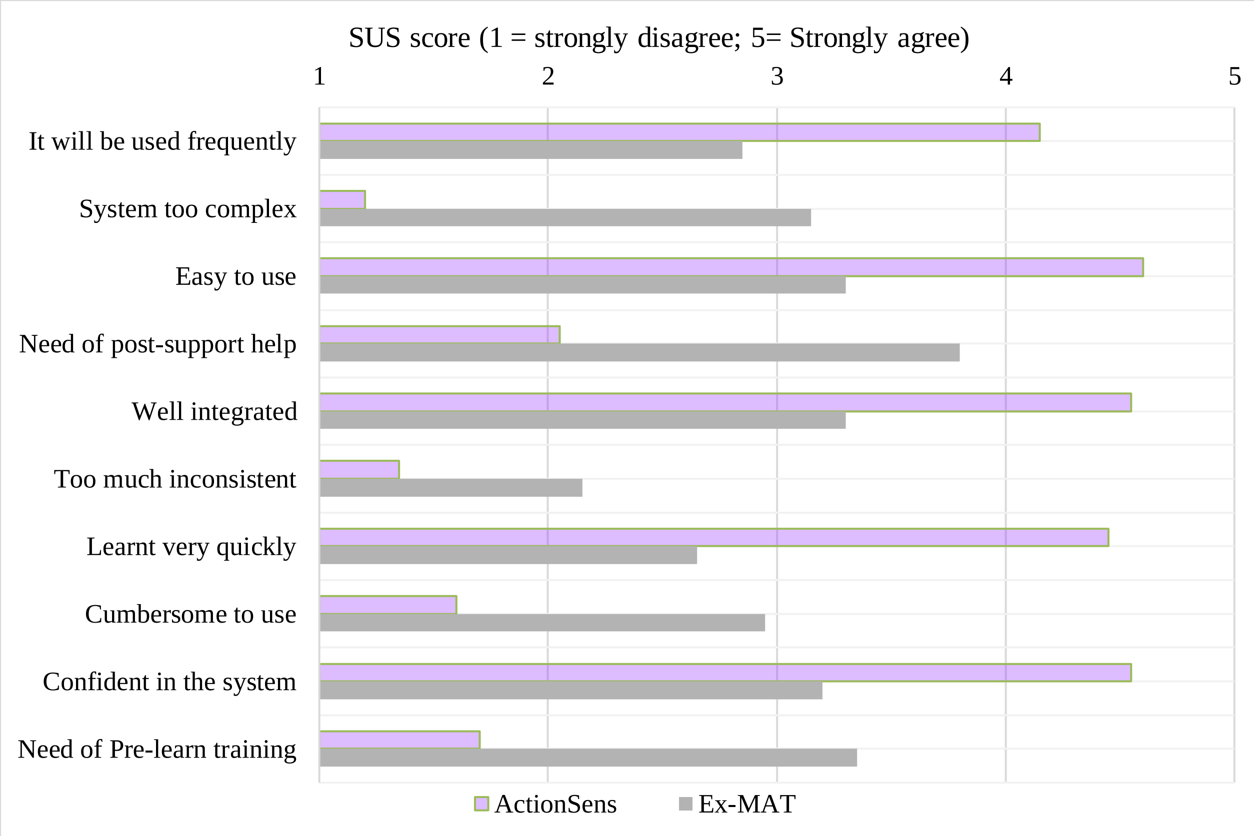


Figure 6: Comparison of SUS sub-scales.

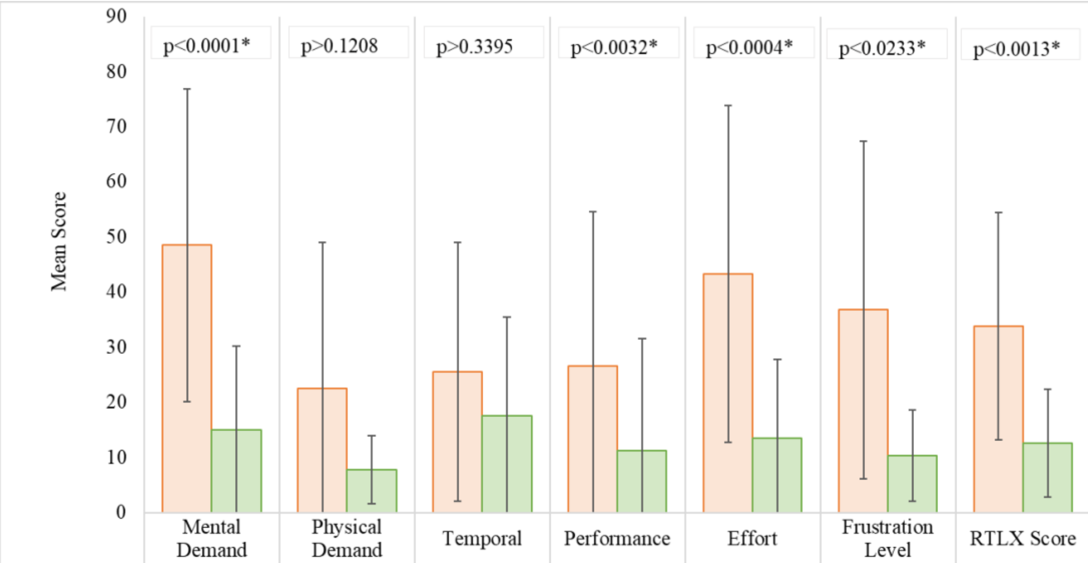
Table 4 presents the results of all the WSRT comparisons conducted between the two conditions. Significant differences (p<0.05) were found in all the tests, indicating statistically significant variations between the groups.

Table 4: Descriptive statistics and WSRT results of usability scores (adjusted).

System Usability Scales	**ActionSens**		**Ex-MAT**		**ActionSens vs Ex-MAT**
	Median	SD	Median	SD	**p-value**
It will be used frequently	3	0.48	2	0.73	<0.0001*
System too complex	4	0.40	1	1.24	<0.0001*
Easy to use	4	0.58	2.5	1.19	0.0005*
Need of post-support help	3	0.97	1	0.81	<0.0001*
Well integrated	4	0.67	2	0.95	0.0001*
Too much inconsistent	4	0.48	2.5	1.01	0.0123*
Learnt very quickly	4	0.74	1	1.28	0.0002*
Cumbersome to use	3.5	0.66	2	1.24	0.0010*
Confident in the system	4	0.67	2	1.21	0.0005*
Need of pre-learn training	3.5	0.84	2	1.11	<0.0001*

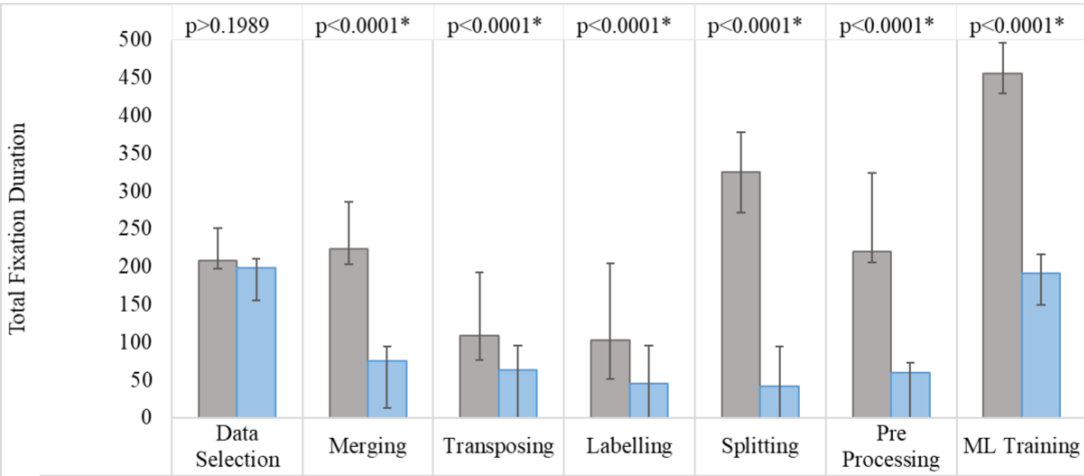
4.2.3 Cognitive workload

The average rating of all participants is depicted by calculating their mean ratings presented in Figure 7 for each of the six subscales and conditions. The final scale illustrates the mean RTLX score for each condition. Figure 7 also shows the WSRT results and whether statistically significant differences (in terms of p-value) existed between the independent variables of the Ex-MAT and ActionSens conditions, where dependent variables were the subscales. Only the cognitive workload of physical demand and temporal demand were not statistically significant.



	Mental Demand	Physical Demand	Temporal	Performance	Effort	Frustration Level	RTLX Score
Ex-MAT	48.5	22.5	25.5	26.5	43.25	36.75	33.833
ActionSens	15	7.75	17.5	11.25	13.5	10.25	12.542

Figure 7: Comparison of perceived workload (NASA-TLX) between the two conditions.



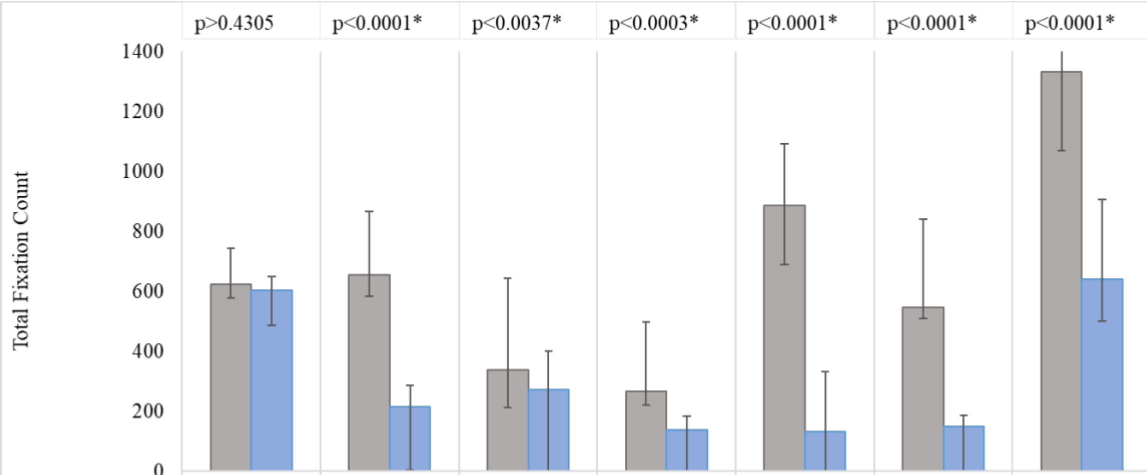
	Data Selection	Merging	Transposing	Labelling	Splitting	Pre Processing	ML Training
Ex-MAT	208.20	222.93	108.99	102.17	324.60	219.47	455.29
ActionSens	198.82	75.38	62.98	45.31	41.46	59.07	190.69

Figure 8: Comparison of total fixation duration mean in specific AOIs for both conditions.

4.2.4 Eye-tracking

The eye-tracking data such as fixation duration, count, visit duration, and count can be directly extracted from Tobii ProLab. Figure 8 presents a total fixation duration (seconds) averaged across all participants for both experimental conditions (Ex-MAT and ActionSens). The total fixation duration indicates participants' cumulative fixation time on the categorized AOIs representative analytics steps. The bar chart illustrates mean differences in fixation duration between the two conditions, with the x-axis representing AOIs and the y-axis representing fixation duration in seconds. The WSRT indicates that there was no statistically significant difference in the total fixation duration in the data selection AOI between both experimental conditions.

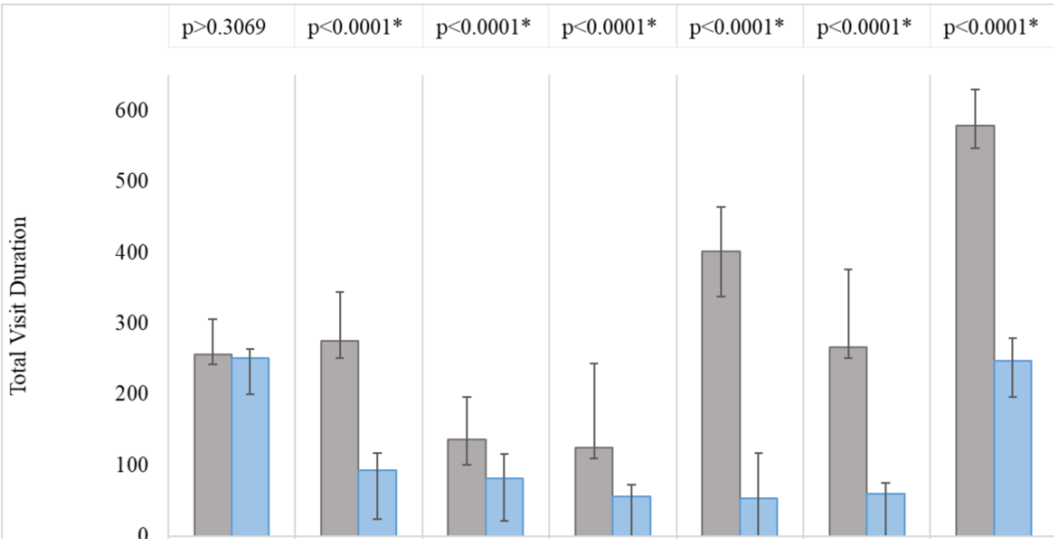
The mean of the total fixation count of the participants on each AOI is presented in Figure 9. This illustrates a comparison of the average number of times the participants fixated on the specific AOIs. The WSRT results of the total fixation count show that there was no statistically significant difference in the data selection AOI between both experimental conditions.



	Data Selection	Merging	Transposing	Labelling	Splitting	Pre Processing	ML Training
Ex-MAT	624.79	654.63	339.58	266.11	887.63	547.79	1332.05
ActionSens	605.11	216.32	272.74	138.00	133.37	149.58	642.32

Figure 9: Comparison of total fixation count mean in specific AOIs.

Figure 10 illustrates a comparison of the mean of the total visit duration (seconds) across all participants, providing insights into the time participants allocated to visiting each AOI. In other words, the duration to process information and complete the task.



	Data Selection	Merging	Transposing	Labelling	Splitting	Pre Processing	ML Training
Ex-MAT	255.68	274.64	135.60	125.12	401.36	266.15	579.08
ActionSens	250.34	93.43	81.83	56.18	53.26	59.87	246.83

Figure 10: Comparison of visit duration mean (seconds) in specific AOIs.

Figure 11 presents the total visit count mean as a comparison of how often the users had visited the specific AOIs in both conditions. The WSRT outcomes about the total count of visits reveal that all the AOIs except for data selection exhibited a statistically significant difference between both experimental conditions.

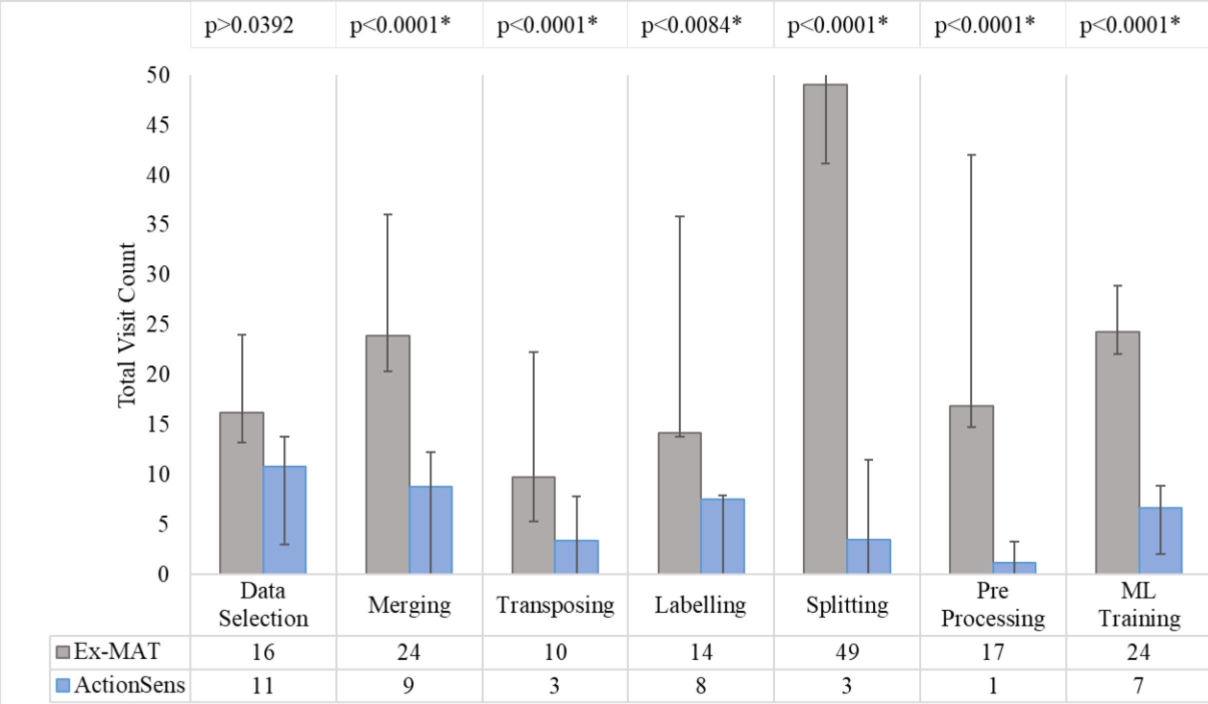


Figure 11: Comparison of total visit count mean in specific AOIs.

4.3 Verbal feedback

After completing the data analytics tasks for each condition, the participants were engaged in a semi-structured interview setting to share their open-ended responses. These responses were then coded using NVIVO *v.14* software, focusing on questions related to the advantages, challenges, and suggestions associated with each condition. The coding process involved categorizing the responses into main themes, where various codes and sub-codes were accumulated. A total of 30 codes were extracted from the Ex-MAT data, while 32 codes were extracted from the ActionSens data (as shown in Table 5). To ensure consistency, the extracted codes were compared with the transcripts (a total of 40 transcripts were reviewed, i.e., 1 transcript each for Ex-MAT and ActionSens platforms from 20 participants).

Table 5: Qualitative analysis results for ActionSens and Ex-MAT.

ActionSens		**Ex-MAT**	
Themes, Codes, and Sub-Codes	**Frequency**	**Themes and Codes**	**Frequency**
Advantages		Advantages	
✓ Interface-specific advantages	39	✓ Ability to process and analyze datasets	11
○ *Easy understanding of blocks represented by their names*	*9*	✓ Consistent workflow	10
○ *Streamlining multiple information streams on a single screen*	*8*	✓ Industry benefits	8

○ *Aesthetic functionalities*	*7*	✓ Pre-defined Excel formula and MATLAB codes	8
○ *Simplified categorization of blocks*	*6*	✓ Easy-to-follow procedural techniques	5
○ *Simplicity of drag and drop feature*	*5*	✓ Positive user-experience	5
○ *Visualization of analytics results*	*4*	✓ Beneficial for construction operation	2
✓ Simple-easy to use	24	✓ Transition from one program to another	2
✓ Intuitive to use	20	✓ Easy to use	1
✓ Efficient procedural techniques with blocks	17	✓ Practical to use	1
✓ Capability of blocks to execute specific operations	13	✓ Simple	1
✓ Preferred block-based interfaces over traditional tools	11		
✓ Satisfactory user experience	11		
✓ Adequacy of blocks to complete tasks	8		
✓ Helpful in learning	8		
✓ Codes are helpful in explaining background actions	7		
✓ Easy to understand	7		
✓ No challenges	7		
✓ Well-integrated	6		
✓ Self-explanatory	4	<u>Challenges</u>	
✓ Confidence	3	✓ Tedious	21
✓ Industry benefits	2	✓ Finding MATLAB as challenging to use	20
✓ Smooth workflow	2	✓ Anti-user-friendly experience	14
<u>Challenges</u>		✓ Confusing workflow	10
✓ Confusing	12	✓ Unfamiliarity with MATLAB	8
✓ Codes	9	✓ Instructions needed	7
✓ Codes are not helpful	8	✓ Complex workflow	6
✓ Hard to understand	6	✓ Prior experience with Excel and MATLAB needed	5
✓ Learning the new interface	6	✓ Hard to understand	4
✓ Unfamiliarity with block-based application	4	✓ Codes	3
✓ Slow ML model training	2	✓ Time-consuming workflow	3

✓ Instructions needed	1	✓ Transition between Excel and MATLAB	3
<u>Suggestions</u>		✓ Specific machine learning steps (train-test)	2
✓ Undocking of code visualization panel	6	✓ Finding Excel as challenging to use	1
✓ Additional visualizations	5	<u>Suggestions</u>	
✓ User customization of panel properties	4	✓ More advanced features needed	6
✓ Help feature	3	✓ Easier condensation of spreadsheet data	3
✓ Bigger block workspace	2	✓ More time is needed to learn	2
✓ Codes are not necessary	2	✓ Easier transition between Excel and MATLAB	1
✓ Highlighting active code segments	1	✓ Reduced repetition in the workflow	1

Following engagement in the Ex-MAT, participants were invited to share their qualitative feedback regarding the challenges, advantages, and suggestions related to the Ex-MAT's impact on the analytics performance process. Participants highlighted salient features of the Ex-MAT that supported the analytics process, '*I felt like it was software doing what it was designed to do. Essentially like it, it was created and coded so that a user could input information. So, I mean, it, it performed as I expected.*'; '*It was an interesting end result to see the matrix matrices that were developed.*'

One common challenge highlighted was the tediousness of the steps involved: *'I think the most challenging part was using MATLAB with a bunch of data. There was a lot of information, a lot of different files that I needed to import, not only on Excel, but at the same time on MATLAB to create the predictions.*' Participants expressed regarding the frequent copying and pasting of data between Excel and MATLAB, stating, '*The step-by-step approach works, but it is definitely inefficient to solve the end goal problem*'; '*it was also just very cumbersome how many different things you had to copy and paste and how many different files you had to make.*'

The participants offered detailed comments, particularly focusing on the potential advanced features that could be incorporated into Ex-MAT. For instance, participants suggested improvements for the Ex-Mat analytics condition, with one stating, '*I think if Excel could just probably merge data... so that you can select other types of files and then merge all of that data in one workspace, that would be great.*' Another participant mentioned the need to optimize the large data sets: '*it was just a lot to do, it was rows and rows of data in Excel. So, if there's a way to like condense that*'.

After conducting analytics performance on ActionSens, participants provided feedback on various salient features specific to the interface, which were categorized under the theme of advantages, challenges, and suggestions of ActionSens. In the advantages, one participant emphasized the benefits of having multiple sources of information displayed on the same screen, stating, '*The information is very organized. You can see. You can organize the data and see how the data is looking. So that was very, very good.*' Another participant expressed a successful understanding of the required actions, stating, '*I understood exactly what they wanted from me, selection, sorting, merging, everything like that.*' Additionally, participants appreciated the clear labels and categorizations, remarking, '*Labels and categorization make sense. It all looks like it's in the right sequence that you need for the block.*' General advantages with the highest frequency covered simplicity or ease of use, intuitiveness, efficiency, platform capability, and preferences of block-based tools over traditional and satisfactory user experience. Some highlighted excerpts from participants: '*It was easy to observe that it would give me data right away. So, if it was giving me data right away means that it just sorting it out.*'; '*That is pretty cool. And like I said, I've never done coding before, so, so seeing something like this and just pulling it over and it like typing in a couple letters or numbers in and it like pops up with his overs pretty nice. It's like self-satisfying*'; '*Pretty well. It takes very few commands for it to do what you want. And the commands you do have to put in are pretty self-explanatory. O yeah, so I would say I would rate it well.*'; '*Like even when you go within the sections, it has everything that you need* *and everything that you would hope is there that performs the exact same thing as it did on MATLAB slash Excel, but important.*'

Participants highlighted two primary challenges they encountered when interacting with the interface of ActionSens. The first challenge was related to the code generator panel, with participants expressing that the codes appeared more complicated compared to the blocks. Participants mentioned, '*The codes look a lot more complicated than the blocks.'; 'Like that's just kind of discouraging.*' The second challenge was related to the initial understanding needed for the block-based environment which led to confusion for some participants. Participants reported that it took some time to grasp the concepts and correctly identify which blocks to use. One participant stated, '*If it was my first time, no one was there. It takes me a minute to pick up, like it would take me a while to pick up, which block can go with which block and just making sure they are the correct ones.*' Another participant expressed difficulty in selecting the appropriate data-related functions within the interface, saying, '*No, I think I just have to play around with the different sections to figure out what they, what they meant. But I think that would take me like shoot a little time.*'

Participants suggested improvements for the ActionSens analytics condition, including additional visualization options and user customization for resizing and repositioning panels. One participant suggested, '*It might be cool if you could full screen the block workspace and then have a minimal view of it, so you can access it whenever needed.'* Another participant proposed, '*Having a feature that turns data into a bar graph or pie chart would be helpful to visually assess how well it matched up.*'

5. DISCUSSION

An experimental study was performed to compare ActionSens, a block-based programming or end-user programming platform to traditional alternatives such as a combination of Excel and MATLAB, or Ex-MAT in the context of performing sensor data analytics by construction students.

5.1 Usability

The results of SUS scores indicate a high level of usability of ActionSens (score = 86 or Grade 'A'), as scores in this range typically correspond to a percentile rank of 90% or above. In contrast, the Ex-MAT obtained a usability score of 49.75, which places it in the lowest category, Grade 'F'. Scores in this range indicate poor usability, with a percentile rank below 50%. This suggests that the majority of participants evaluated ActionSens as significantly more usable compared to Ex-MAT. This also implies that all participants evaluated the two conditions differently, so the typical difference in the usability of the two systems was subjected to further decomposition to understand where they differed. Further decomposed items of the SUS showed that all the perceived usability item variations were statistically significant. First, participants strongly agreed that they would like to use ActionSens more frequently (p-value<0.0001) and it was easier to use (p-value=0.0005), and the platform had more well-integrated functions (p-value=0.0001) indicating a positive attitude and seamless user experience. The significance levels suggest that this preference was significantly different from their perception of Ex-MAT. One participant commented, '*Overall interface I felt was simple, easy to guide, easy to educate you. Pretty self-explanatory if you ever going to open it up yourself and try to do it yourself.*' Participants appreciated the simplicity of the block diagram, which made it accessible even to those with no prior coding or computer experience as mentioned, '*It's one of the easiest forms of programming that I think I've ever had to deal with.*'; '*... the block diagram is simpler, like simple enough for someone to be able, which completely fresh doesn't know anything about coding or even like someone who has no experience in computer.*'

Additionally, participants felt more confident (p-value=0.0005) in using ActionSens and perceived that they learned how to use it more quickly (p-value= 0.0002). This is highlighted by the feasibility of the analytics workflow and the presentation of organized information on a single screen with color-coded puzzle shapes were perceived as aesthetically pleasing and user-friendly. One participant emphasized this in this statement, '*I like that everything is color coded. It made it a lot easier to use so I appreciated that.'; 'I like the, like shapes. It's very obvious of what goes where. And like the linear reading of it was also helpful to see.*'; '*I do like the fact that the shapes align together. Just literally puzzle pieces, and the colors are good too, just to keep stuff separate from each other. Cause if they were all the same color, it probably would be more confusing.*' Moreover, participants noted that the block-based approach of ActionSens allowed for a more accelerated option in teaching data analysis to construction students. One participant remarked, '*Whereas with the block-based software, you could spend a week,* *maybe two, two lectures on it, just understanding the concept and then how to utilize this. It would be more of an accelerated option for teaching the students how to properly analyze their data and as well as practically used.*'

In contrast, Ex-MAT did not evoke the same confidence or quick learnability. Taken together, the rest of the itemized findings highlight the specific areas where the Ex-MAT fell short in terms of usability when compared to ActionSens. The system was perceived as more unnecessarily complex (p-value<0.0001), requiring technical support (p-value<0.0001), exhibiting inconsistency (p-value=0.0123), being cumbersome to use (p-value=0.0010), and demanding a steep learning curve (p-value<0.0001). These perceptions of the Ex-MAT could have made it more challenging for users to understand and navigate the system effectively and to acquire the necessary knowledge and skills to operate it proficiently. Overall, these findings demonstrate that ActionSens outperformed Ex-MAT in terms of user preference, confidence, ease of use, effective integration of functionalities, and learnability. This study is corroborated by results from past research studies that have repeatedly shown the block-based paradigm's fair user-friendliness, which has produced favorable assessments in terms of usability (Dawoud et al., 2021), usefulness, user satisfaction (Calderon et al., 2022), and quick learnability (Rough, 2018). For instance, Calderon et al. (2022) used Google Blocky to provide visual programming of the algorithms. Rough (2018) noted that blocks-based languages are suitable for particular problem domains, where the domain-specific terms can be mapped directly into block representations. This helped improve the learning experience of learners as shown in previous studies (Glas et al., 2022, Barboza et al., 2023, Mahadevan et al., 2016).

5.2 Cognitive Load

NASA-TLX questionnaires were used to measure users' cognitive load during the execution of the data analytics tasks using both ActionSens and Ex-MAT. First, the overall cognitive workload of users with ActionSens was comparatively lower than expected. The results show that participants perceived a significantly higher workload in terms of mental demand (p-value<0.0001), performance (p-value=0.0032), effort (p-value=0.0004), frustration level (p-value=0.0233), and overall raw TLX score (p-value=0.0003) in the Ex-MAT condition compared to the ActionSens condition. However, there was no significant difference in perceived workload in terms of physical demand between the two conditions. This is expected, as the learners did not require any physical aspects while using both platforms. The results of NASA-TLX can be reinforced by participants' qualitative assessment, which highlights the lower likelihood of errors leading to the desired outcome with less effort. As one participant put it, '*You're not necessarily going to be able to mess it up as easy as you would if you forgot a semicolon or forward slash as you would in standard coding, but you're still able to get the desired outcome with a lot less effort.*' Additionally, another participant emphasized the mental state induced by observing the complete experience of the ActionSens condition, expressing, '*I had an excitement, I had a very pleasant feeling of joy that it did accomplish what I wanted. It accomplished it in the way that I. I'm very organized person. So, seeing this organization itself makes me very, very, very comfortable, very joyful, and gives me no frustration or anger at all because it's very organized, simplicity and it just step by step in a way that just makes sense.*' The findings are consistent with prior studies showing that block-based platforms lead to reduced cognitive load compared to alternative text-based languages(Pratidhina et al., 2021, Glas et al., 2023). This is because Blockly offers the advantage of being used for unfamiliar tasks and particularly complex tasks such as sensor data analytics (Glas et al., 2022). Similarly, the characteristics of block-based languages which eliminate syntactic errors and ensure that users only recognize useful blocks needed to solve a problem help reduce users' working memory demand (Tulving, 1985).

5.3 Visual Attention and Impact on Overall User Experience

This eye-tracking analysis highlights the nuanced differences in eye-movement patterns between the two groups (i.e., users of ActionSens and Ex-MAT platforms). Overall, all the fixation duration, counts, visit duration, and counts were lesser for ActionSens, compared to Ex-MAT, across all the 7 key tasks or AOIs. Initially, in the first task of the ActionSens workflow (i.e., data selection), both the total fixation duration and visit durations were longer, almost close to Ex-MAT (**Error! Reference source not found.** and **Error! Reference source not found.**). However, in ActionSens, there was a decline in these durations in subsequent steps, followed by a noticeable increase in the last step, ML training. Despite this increase, the total fixation duration and visit durations remained significantly lower (p-value<0.05) than those observed in Ex-MAT throughout the rest of its workflow. A similar trend was observed in the comparison of means and the statistically significant difference in both total fixation counts and visit counts (**Error! Reference source not found.** and **Error! Reference source not found.**).

In this study, the tasks performed by participants in both situations were essentially identical in terms of objectives, aiming to achieve comparable outcomes. However, the observed differences in eye-tracking patterns can affect the constrained capacity of working memory, which is directly linked to visual attentional resources and consequently impacts the perceived cognitive load and usability. Fixation duration and fixation count are measures that reflect visual attentional resources (De Koning et al., 2010). In that regard, Ex-MAT condition required participants to devote longer durations (fixation and visit durations) and a greater number of steps (fixation and visit counts) to complete the tasks, which demanded more mental effort or attentional resources and dissatisfaction with the interface usability potentially increasing participants' perceived cognitive load and negatively impacting usability. Additionally, it may be inferred that the participants found information processing on the Ex-MAT condition complex because longer fixation length is associated with complexity, increased cognitive processing, and difficulty in information extraction (Pan et al., 2004, Pachman et al., 2016). A higher number of task steps (switching between multiple windows and platforms) can be related to fluctuations of attentional states, thus involving an increase in attentional processing and cognitive load (Di Stasi et al., 2011). Furthermore, it is well-recognized in HCI studies that time is an important factor that may affect perceived fatigue (Käthner et al., 2014), which could have led to participants perceiving a higher cognitive load and negatively impacting the overall usability of Ex-MAT.

In contrast, the ActionSens platform provided a unified interface for accessing all the necessary information (such as blocks menu, codes, analytics visualizer, and video playback), eliminating the need for frequent application switching as required in the Ex-MAT condition. This resulted in a shorter duration of overall fixations and visits in ActionSens. This shows that participants needed less time to process what they viewed. This streamlined interface design in ActionSens may have better-sustained participants' visual attentional resources, ultimately contributing to a lower cognitive load and a higher usability score. This situation offers the opportunity to decrease visual complexity and enhance the chances of improving visual search efficiency, reducing cognitive load, and facilitating essential cognitive processing (De Koning et al., 2010). Furthermore, less time consumed for any task as a measure of usability could indicate increased work efficiency and ease of learning, ultimately resulting in improved productivity (Punchoojit and Hongwarittorrn, 2017). A particular participant's statement highlights some salient features of ActionSens that support their positive feedback. The participant appreciated the efficient use of the interface workspace, mentioning, '*And there's enough space to get everything done. There's really no wasted space that could be utilized.'* In addition, the clear labeling and organization of ActionSens were highlighted by a participant who mentioned, '*So, it's, it's also labeled pretty well organized pretty well. I said this last. I'll say it again.*' This perception is supported by research on the design of block-based programming interfaces to improve learner satisfaction and usability without affecting their performance (Rodríguez et al., 2017). This study compared three layouts of block categorization. The functionality interface, with reduced categories and inspired by 'Control,' 'Operators,' and 'Input/Output,' or simply categorized based on functions aiming to simplify navigation and reduce domain vocabulary knowledge required to use the interface. Results indicated that this version achieved the highest usability score and user satisfaction compared to the other conditions that had no categories at all or used Blockly's default categories ('Logic,' 'Loops,' 'Math,' and 'Text').

6. CONCLUSIONS, LIMITATIONS AND FUTURE WORKS

This article presents an experimental study of an end-user programming platform for equipping construction students with sensor data analytics skills. The study evaluates participants' objective indicators and subjective perceptions to explore how the tool facilitates sensor data analytics through computational thinking methods in construction education. ActionSens turned out to be a more user-friendly technology that enabled users to perform data analytics with a more manageable cognitive load and visual attentional resources than the contrasted Ex-MAT. The advantages of ActionSens as an analytics platform for the intended audience (i.e., construction students) were demonstrated by the assessment metrics utilized in this study. This performance advantage is consistent with the findings from the SUS, NASA-TLX, and relevant eye-tracking data. It can be inferred that the visual efficiency and user-friendliness of the ActionSens platform contributed to construction students perceiving sensor data analytics, using machine-learning techniques, as more feasible compared to Ex-MAT. This aligns with the main objective of the platform, which aims to interactively guide students in performing sensor data analytics. In doing so, this study builds on prior literature which suggests that EUP tools can be an effective way to support the development of domain-specific and CT skills, while also contributing to the theoretical frameworks for technological learning platforms and cognitive load (i.e., LfU and CLT). The results present opportunities for construction educational practitioners and UX experts to develop resources that are specifically tailored to improve engagement and learning outcomes in construction education. By taking into account cognitive, usability, and attentional aspects, the creation and integration of pedagogical EUP platforms in various levels of construction education can effectively bridge knowledge gaps and adequately equip the workforce to meet the increasing technological skill requirements. Furthermore, this study signifies the potential for integrating authentic data-driven methods into block-based programming environments for construction-related data analytics. Unlike traditional assessment designs that rely on contrived scenarios, the approach in this study involves developing models that derive characteristics from actual construction data samples, enabling users to engage with authentic datasets. This utilization of authentic data instills student confidence in their ability to effectively analyze data for goal-oriented task performance, aligning with the principles of the Learning for Use (LfU) theory.

It is necessary to note a few limitations of this study. First, because the EUP platform was customized for a particular construction activity analysis, our findings might not be generalizable to significantly different types of data analytics tasks (i.e., more complex, or open-ended data analytics). Even though the study indicated that ActionSens was founded on developing students' CT and sensor data analytics skills, how the enhancement occurs within interaction was not explored in this study. In forthcoming studies, alternative data analysis methods can be employed to complement the subjective and objective metrics used in this experimental research. By leveraging interaction analytics to examine eye movements within the EUP platform, in-depth insights can be gained regarding dynamic interaction and scaffolding (Tawfik et al., 2022) As a result, researchers will be able to uncover how learners advance in data analytics through their CT processes or skills by analyzing user visual search patterns. Additionally, this would enable researchers to pinpoint the 'when' and 'where' people engage in aid-seeking behavior and further triangulate these findings using qualitative information (Tawfik et al., 2022). Besides, future research could entail exploring the gathered data from the perspective of individual differences to gain insights into how users with diverse backgrounds (such as gender, age, program, programming, analytics, and internship experiences) perceive and engage with the EUP platform. Additional physiological sensing techniques, such as EEG data, can be utilized in training classification models to predict the cognitive states of the users.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) [grant numbers 2111003 and 2111045].

REFERENCES

Adepoju, O. O. & Aigbavboa, C. O. 2021. Assessing knowledge and skills gap for construction 4.0 in a developing economy. Journal of Public Affairs, 21, e2264.

Aggarwal, C. C. An Introduction to Sensor Data Analytics. Managing and Mining Sensor Data, 2013.

Akanmu, A. A., Akligo, V. S., Ogunseiju, O. R., Lee, S. W. & Murzi, H. 2022. Data Analytics and Computational Thinking Skills in Construction Engineering and Management Education: A Conceptual System. 204-213.

Akhavian, R. & Behzadan, A. H. 2015. Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers. Advanced Engineering Informatics, 29, 867-877.

Almusaly, I., Metoyer, R. & Jensen, C. Evaluation of A Visual Programming Keyboard on Touchscreen Devices. 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 1-4 Oct. 2018 2018. 57-64.

Anumba, C. J., Akanmu, A., Yuan, X. & Kan, C. 2021. Cyber—physical systems development for construction applications. Frontiers of Engineering Management, 8, 72-87.

Arabshahi, M., Wang, D., Sun, J., Rahnamayiezekavat, P., Tang, W., Wang, Y. & Wang, X. 2021. Review on Sensing Technology Adoption in the Construction Industry. Sensors (Basel, Switzerland), 21, 8307.

Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., Shringi, A. & Mendis, P. 2022. Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. Automation in Construction, 141, 104440.

Barboza, L., Mello, R., Modell, M. & Teixeira, E. S. Blockly-DS: Blocks Programming for Data Science with Visual, Statistical, Descriptive and Predictive Analysis. LAK23: 13th International Learning Analytics and Knowledge Conference, 2023. 644-649.

Barral, O., Lallé, S., Guz, G., Iranpour, A. & Conati, C. Eye-tracking to predict user cognitive abilities and performance for user-adaptive narrative visualizations. Proceedings of the 2020 International Conference on Multimodal Interaction, 2020. 163-173.

Barricelli, B., Fogli, D. & Locoro, A. 2023. EUDability: A new construct at the intersection of End-User Development and Computational Thinking. Journal of Systems and Software, 195, 111516.

Barricelli, B. R., Cassano, F., Fogli, D. & Piccinno, A. 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. Journal of Systems and Software, 149, 101-137.

Bau, D., Gray, J., Kelleher, C., Sheldon, J. & Turbak, F. 2017. Learnable programming: blocks and beyond. Communications of the ACM, 60, 72-80.

Berti, S., Paterno, F. & Santoro, C. 2006. Natural development of nomadic interfaces based on conceptual descriptions. End user development. Springer.

Bojko, A. Eye tracking in user experience testing: How to make the most of it. Proceedings of the UPA 2005 Conference, 2005.

Borys, M. & Plechawska-Wójcik, M. 2017a. Eye-tracking metrics in perception and visual attention research. EJMT, 3, 11-23.

Borys, M. & Plechawska-Wójcik, M. Eye-tracking metrics in perception and visual attention research. 2017b.

Brunyé, T. T., Drew, T., Weaver, D. L. & Elmore, J. G. 2019. A review of eye tracking for understanding and improving diagnostic interpretation. Cognitive research: principles and implications, 4, 1-16.

Calderon, J. F., Rojas, L. A., Sorbello, K. & Acero, N. User Experience Evaluation of a Computational Thinking-Enhanced Problem-Solving Tool: Findings and Next Steps. In: MEISELWITZ, G., ed. Social Computing and Social Media: Design, User Experience and Impact, 2022// 2022 Cham. Springer International Publishing, 13-27.

Chen, T.-L., Chen, Y.-R., Yu, M.-S. & Lee, J.-K. 2021. NNBlocks: a Blockly framework for AI computing. The Journal of Supercomputing, 77, 8622-8652.

Cheng, T., Teizer, J., Migliaccio, G. C. & Gatti, U. C. 2013. Automated task-level activity analysis through fusion of real time location sensors and worker's thoracic posture data. Automation in Construction, 29, 24-39.

Coronado, E., Deuff, D., Carreno-Medrano, P., Tian, L., Kulić, D., Sumartojo, S., Mastrogiovanni, F. & Venture, G. 2021. Towards a modular and distributed end-user development framework for human-robot interaction. IEEE Access, 9, 12675-12692.

D'alba, A. & Huett, K. C. 2017. Learning Computational Skills in uCode@UWG: Challenges and Recommendations. In: RICH, P. J. & HODGES, C. B. (eds.) Emerging Research, Practice, and Policy on Computational Thinking. Cham: Springer International Publishing.

Dawoud, F., Adel, A. & Sharaf, N. 2021. Collaborative Coding in a Robotic Visual Language.

De Koning, B. B., Tabbers, H. K., Rikers, R. M. & Paas, F. 2010. Attention guidance in learning from a complex animation: Seeing is understanding? Learning and instruction, 20, 111-122.

Derisma, D. 2020. The usability analysis online learning site for supporting computer programming course using system usability scale (SUS) in a university.

Di Stasi, L. L., Antolí, A. & Cañas, J. J. 2011. Main sequence: An index for detecting mental workload variation in complex tasks. Applied Ergonomics, 42, 807-813.

Edelson, D. 2001a. Learning-for-Use: A Framework for the Design of Technology-Supported Inquiry Activities. Journal of Research in Science Teaching, 38, 355-385.

Edelson, D. C. 2001b. Learning-for-use: A framework for the design of technology-supported inquiry activities. Journal of Research in Science teaching, 38, 355-385.

Ehmke, C. & Wilson, S. 2007. Identifying web usability problems from eyetracking data.

Ellis, G. 2020. Sensing the Future of Building: The Role of Sensors in Construction [Online]. Available: https://constructionblog.autodesk.com/sensors-in-construction/ [Accessed].

Galan, D., Heradio, R., De La Torre, L., Dormido, S. & Esquembre, F. 2017. Conducting online lab experiments with Blockly. IFAC-PapersOnLine, 50, 13474-13479.

Georgsson, M., Staggers, N., Årsand, E. & Kushniruk, A. 2019. Employing a user-centered cognitive walkthrough to evaluate a mHealth diabetes self-management application: A case study and beginning method validation. Journal of Biomedical Informatics, 91, 103110.

Glas, M., Vielberth, M., Reittinger, T., Böhm, F. & Pernul, G. Visual Programming in Cyber Range Training to Improve Skill Development. In: CLARKE, N. & FURNELL, S., eds. Human Aspects of Information Security and Assurance, 2022 Cham. Springer International Publishing, 3-13.

Glas, M., Vielberth, M., Reittinger, T., Böhm, F. & Pernul, G. 2023. Improving cybersecurity skill development through visual programming. Information & Computer Security, ahead-of-print.

Goldberg, J. H. & Kotval, X. P. 1999. Computer interface evaluation using eye movements: methods and constructs. International journal of industrial ergonomics, 24, 631-645.

Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N. & Wichansky, A. M. Eye tracking in web search tasks: design implications. Proceedings of the 2002 symposium on Eye tracking research & applications, 2002. 51-58.

Gonsalves, N., Ogunseiju, O. R. & Akanmu, A. A. 2022. Activity recognition from trunk muscle activations for wearable and non-wearable robot conditions. Smart and Sustainable Built Environment.

Google Inc. 2020. Blockly Developers Google [Online]. Available: https://developers.google.com/blockly [Accessed].

Gupta, V., Irimia, J., Pau, I. & Rodríguez-Patón, A. 2017. BioBlocks: Programming Protocols in Biology Made Easier. ACS Synthetic Biology, 6, 1230-1232.

Hart, S. G. & Staveland, L. E. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Advances in psychology. Elsevier.

Hartson, R. & Pyla, P. S. 2012. The UX Book: Process and guidelines for ensuring a quality user experience, Elsevier.

Heureux, A. L., Grolinger, K., Higashino, W. A. & Capretz, M. a. M. A Gamification Framework for Sensor Data Analytics. 2017 IEEE International Congress on Internet of Things (ICIOT), 25-30 June 2017 2017. 74-81.

Hsieh, H.-F. & Shannon, S. E. 2005. Three Approaches to Qualitative Content Analysis. Qualitative Health Research, 15, 1277-1288.

Jacob, R. J. & Karn, K. S. 2003. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. The mind's eye. Elsevier.

Karakasis, C. & Xinogalos, S. 2020. BlocklyScript: Design and pilot evaluation of an RPG platform game for cultivating computational thinking skills to young students. Informatics in Education, 19, 641-668.

Karat, J. User-Centered Software Evaluation Methodologies. 1997.

Käthner, I., Wriessnegger, S. C., Müller-Putz, G. R., Kübler, A. & Halder, S. 2014. Effects of mental workload and fatigue on the P300, alpha and theta band power during operation of an ERP (P300) brain–computer interface. Biological psychology, 102, 118-129.

Kensing, F. & Munk-Madsen, A. 1993. PD: Structure in the Toolbox. Commun. ACM, 36, 78-85.

Khalid, M., Akanmu, A., Murzi, H., Lee, S. W., Awolusi, I., Manesh, D. & Okonkwo, C. 2023. Industry Perception of the Knowledge and Skills required to Implement Sensor Data Analytics in Construction. Journal of Civil Engineering Education, 150.

Korbach, A., Brünken, R. & Park, B. 2016. Learner characteristics and information processing in multimedia learning: A moderated mediation of the seductive details effect. Learning and Individual Differences, 51, 59-68.

Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A. & Qureshi, B. 2020. An overview of IoT sensor data processing, fusion, and analysis techniques. Sensors, 20, 6076.

Kumar, N. & Kumar, J. 2016. Measurement of Cognitive Load in HCI Systems Using EEG Power Spectrum: An Experimental Study. Procedia Computer Science, 84, 70-78.

Lai, M.-L., Tsai, M.-J., Yang, F.-Y., Hsu, C.-Y., Liu, T.-C., Lee, S. W.-Y., Lee, M.-H., Chiou, G.-L., Liang, J.-C. & Tsai, C.-C. 2013. A review of using eye-tracking technology in exploring learning from 2000 to 2012. Educational research review, 10, 90-115.

Lee, J. Y., Donkers, J., Jarodzka, H. & Van Merriënboer, J. J. G. 2019. How prior knowledge affects problem-solving performance in a medical simulation game: Using game-logs and eye-tracking. Computers in Human Behavior, 99, 268-277.

Lewis, J. R. 1994. Sample sizes for usability studies: Additional considerations. Human factors, 36, 368-378.

Liu, J., Luo, H. & Liu, H. 2022. Deep learning-based data analytics for safety in construction. Automation in Construction, 140, 104302.

Lovasz-Bukvova, H., Hölzl, M., Kormann-Hainzl, G., Moser, T., Zigart, T. & Schlund, S. Usability and Task Load of Applications in Augmented and Virtual Reality. In: YILMAZ, M., CLARKE, P., MESSNARZ, R. & REINER, M., eds. Systems, Software and Services Process Improvement, 2021// 2021 Cham. Springer International Publishing, 708-718.

Mahadevan, A., Freeman, J. & Magerko, B. An interactive, graphical coding environment for EarSketch online using Blockly and Web Audio API. Proceedings of the 2nd Web Audio Conference (WAC'16), 2016.

Mansouri, S., Castronovo, F. & Akhavian, R. 2020. Analysis of the Synergistic Effect of Data Analytics and Technology Trends in the AEC/FM Industry. Journal of Construction Engineering and Management, 146, 04019113.

Martín, H., Bernardos, A. M., Iglesias, J. & Casar, J. R. 2013. Activity logging using lightweight classification techniques in mobile devices. Personal and ubiquitous computing, 17, 675-695.

Miles, M. B., Huberman, A. M. & Saldaña, J. 2018. Qualitative data analysis: A methods sourcebook, Sage publications.

Ngo, J., Hwang, B.-G. & Zhang, C. 2020. Factor-based big data and predictive analytics capability assessment tool for the construction industry. Automation in Construction, 110, 103042.

Ogunseiju, O., Akanmu, A. & Bairaktarova, D. 2021. Sensing Technologies in Construction Engineering and Management Programs: A Comparison of Industry Expectations and Faculty Perceptions. Proceedings of 57th Associated Schools of Construction Conference.

Olney, A. M. & Fleming, S. D. A Cognitive Load Perspective on the Design of Blocks Languages for Data Science. 2019 IEEE Blocks and Beyond Workshop (B&B), 18-18 Oct. 2019 2019. 95-97.

Paas, F. & Sweller, J. 2014. Implications of Cognitive Load Theory for Multimedia Learning. In: MAYER, R. E. (ed.) The Cambridge Handbook of Multimedia Learning. 2 ed. Cambridge: Cambridge University Press.

Pachman, M., Arguel, A., Lockyer, L., Kennedy, G. & Lodge, J. 2016. Eye tracking and early detection of confusion in digital learning environments: Proof of concept. Australasian Journal of Educational Technology, 32.

Pan, B., Hembrooke, H. A., Gay, G. K., Granka, L. A., Feusner, M. K. & Newman, J. K. The determinants of web page viewing behavior: an eye-tracking study. Proceedings of the 2004 symposium on Eye tracking research & applications, 2004. 147-154.

Park, B., Knörzer, L., Plass, J. L. & Brünken, R. 2015. Emotional design and positive emotions in multimedia learning: An eyetracking study on the use of anthropomorphisms. Computers & Education, 86, 30-42.

Pernice, K. & Nielsen, J. 2009. How to conduct eyetracking studies. Nielsen Norman Group, 945397498.

Poole, A. & Ball, L. J. 2006. Eye tracking in HCI and usability research. Encyclopedia of human computer interaction. IGI global.

Pratidhina, E., Rosana, D., Kuswanto, H. & Dwandaru, W. S. B. 2021. Using Arduino and online block-structured programing language for physics practical work. Physics Education, 56, 055028.

Punchoojit, L. & Hongwarittorrn, N. 2017. Usability Studies on Mobile User Interface Design Patterns: A Systematic Literature Review. Advances in Human-Computer Interaction, 2017, 6787504.

Rahaman, M. M., Mahfuj, E., Haque, M. M., Shekdar, R. S. & Islam, K. Z. 2020. Educational Robot for Learning Programming through Blockly based Mobile Application. Journal of Technological Science & Engineering (JTSE), 1, 21-25.

Ramoğlu, M., Genç, Ç. & Rızvanoğlu, K. Programming a Robotic Toy with a Block Coding Application: A Usability Study with Non-programmer Adults. In: MARCUS, A. & WANG, W., eds. Design, User Experience, and Usability: Theory, Methodology, and Management, 2017// 2017 Cham. Springer International Publishing, 652-666.

Rashid, K. M. & Louis, J. Construction equipment activity recognition from IMUs mounted on articulated implements and supervised classification. ASCE International Conference on Computing in Civil Engineering 2019, 2019. American Society of Civil Engineers Reston, VA, 130-138.

Rijo-García, S., Segredo, E. & León, C. 2022. Computational thinking and user interfaces: A systematic review. IEEE Transactions on Education.

Robson, C. & Mccartan, K. 2016. Real world research, Wiley Global Education.

Rodríguez, F. J., Price, K. M., Isaac, J., Boyer, K. E. & Gardner-Mccune, C. How block categories affect learner satisfaction with a block-based programming interface. 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 11-14 Oct. 2017 2017. 201-205.

Rosson, M. B. & Carroll, J. M. 2002. Usability engineering: scenario-based development of human-computer interaction, Morgan Kaufmann.

Rough, D. J. 2018. Jeeves: a blocks-based approach to end-user development of experience sampling apps. University of St Andrews.

Rowe, S., Riggio, M., De Amicis, R. & Rowe, S. R. 2020. Teacher perceptions of training and pedagogical value of cross-reality and sensor data from smart buildings. Education Sciences, 10, 234.

Saldaña, J. 2009. The Coding Manual for Qualitative Researchers, Sage.

Sarmento, H. R., Reis, C. a. S., Zaramella, V., Almeida, L. D. A. & Tacla, C. A. Supporting the Development of Computational Thinking: A Robotic Platform Controlled by Smartphone. In: ZAPHIRIS, P. & IOANNOU, A., eds. Learning and Collaboration Technologies, 2015// 2015 Cham. Springer International Publishing, 124-135.

Sauro, J. 2011. Measuring Usability with the System Usability Scale (SUS). [Online]. Available: https://measuringu.com/sus/ [Accessed].

Shen, X. & Lu, M. 2012. A framework for indoor construction resources tracking by applying wireless sensor networks. Canadian Journal of Civil Engineering, 39.

Sherafat, B., Ahn, C. R., Akhavian, R., Behzadan, A. H., Golparvar-Fard, M., Kim, H., Lee, Y.-C., Rashidi, A. & Azar, E. R. 2020. Automated Methods for Activity Recognition of Construction Workers and Equipment: State-of-the-Art Review. Journal of Construction Engineering and Management, 146, 03120002.

Shute, V. J., Sun, C. & Asbell-Clarke, J. 2017. Demystifying computational thinking. Educational Research Review, 22, 142-158.

Sweller, J. 1988. Cognitive load during problem solving: Effects on learning. Cognitive science, 12, 257-285.

Tawfik, A. A., Payne, L. & Olney, A. M. 2022. Scaffolding Computational Thinking Through Block Coding: A Learner Experience Design Study. Technology, Knowledge and Learning.

Tudoreanu, M. E. 2003. Designing effective program visualization tools for reducing user's cognitive effort. Proceedings of the 2003 ACM symposium on Software visualization. San Diego, California: Association for Computing Machinery.

Tulving, E. 1985. How many memory systems are there? American psychologist, 40, 385.

Van Orden, K. F., Limbert, W., Makeig, S. & Jung, T.-P. 2001. Eye activity correlates of workload during a visuospatial memory task. Human factors, 43, 111-121.

Van Orden, K. F., Nugent, W., La Fleur, B. & Moncho, S. 1998. Assessment of variable coded symbology using visual search performance and eye fixation measures. NAVAL HEALTH RESEARCH CENTER SAN DIEGO CA.

Virzi, R. A. 1992. Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough? Human Factors, 34, 457-468.

Wang, Q., Yang, S., Liu, M., Cao, Z. & Ma, Q. 2014. An eye-tracking study of website complexity from cognitive load perspective. Decision Support Systems, 62, 1-10.

Zhong, Y. iVirtualWorld : A Domain-Oriented End-User Development Environment for Building 3D Virtual Chemistry Experiments. 2013.