ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.5:004.93, 004.42, 004.62

doi: 10.32620/reks.2024.3.03

Serhii ZELINSKYI, Yuriy BOYKO

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

INTEGRATING SESSION RECORDING AND EYE-TRACKING: DEVELOPMENT AND EVALUATION OF A CHROME EXTENSION FOR USER BEHAVIOR ANALYSIS

*The **subject of this** article is the development, implementation, and evaluation of a Chrome extension designed to record and replay web sessions using integrated eye-tracking data. The **goal** is to develop a tool that enhances user interaction analysis by combining session recordings with eye-tracking capabilities. The **tasks** to be solved are as follows: create a functional Chrome extension that utilizes rrweb for session recording and WebGazer.js for eye tracking; implement features, such as session recording, replay with eye-tracking data overlays, and session management; and provide options for exporting and importing recorded sessions. The **methods** used are: architectural modeling using UML diagrams to design the system architecture, software engineering techniques for developing the extension, integration testing to ensure the smooth operation of combined features, and data preprocessing techniques to prevent redundancy and reduce noise in eye-tracking data. Additionally, a structured user study with detailed questionnaires combining both Likert-scale questions and open-ended responses and feedback analysis were conducted to evaluate usability and gather feedback. The following **results** were obtained: the extension was successfully developed and evaluated with 25 participants aged 18–35 years in a controlled environment. High usability ratings were obtained, with an average score of 4.5 out of 5 for the session recording, replay, and session management features. However, the eye-tracking feature received a lower rating of 3.8 out of 5 due to occasional inaccuracies in the eye-tracking data. The qualitative feedback indicated the usefulness of the eye-tracking feature and highlighted the need for improved data accuracy. **Conclusions.** The scientific novelty of this study lies in the integration of session recording and eye tracking within a Chrome extension, which represents a novel and comprehensive tool for user interaction analysis on the web. The tool's ability to capture both behavioral data and visual attention without requiring website code modifications is particularly valuable for researchers, marketing specialists, UI/UX designers, and product developers. The usability study and feedback analysis provided a clear direction for future improvements, including enhancing the eye-tracking accuracy and integrating advanced analytics and customizable reporting options. **Future work** will also explore the integration of machine learning algorithms to automatically analyze recorded data to provide deeper insights and actionable recommendations.*

Keywords: *web session recording; eye-tracking; web session replay; behavioral analysis; usability testing.*

1. Introduction

1.1. Motivation

The increasing complexity of web interfaces and increasing focus on enhancing user experience have created a need for advanced tools that can capture and analyze user interaction data with greater precision. Traditional user behavior analysis methods, such as clickstream data, user surveys, and even modern web analytics tools, often cannot provide a detailed understanding of how users interact with interfaces [1 - 3]. These methods are limited in their ability to correlate user actions with visual attention, which leaves gaps in the analysis of user behavior.

There are several tools available today that offer session recording and interaction analysis, such as Hotjar and FullStory. These tools are effective for capturing and replaying user sessions and provide valuable user behavior insights. However, they do not have the ability to directly measure and analyze where users are looking on a webpage because they do not support eye-tracking. Eye-tracking technology provides a unique perspective on user behavior by revealing visual attention patterns, which are crucial for understanding user intent and decision-making processes [4], identifying usability issues [5, 6], and even enhancing the cybersecurity of computer systems [7].

The motivation for this study was to develop an integrated cross-platform tool that combines session recording with eye-tracking functionality. Such a tool would allow researchers, UI/UX designers, and product developers to gain deeper understanding of user interactions on the web. By offering a unified platform that captures and replays user sessions using synchronized eye-tracking data, this solution aims to enhance our ability to identify usability issues and optimize interface designs.

Through the development of this tool, this research contributes to web usability studies and user behavior analysis. In addition, it opens opportunities for future advancements, such as the integration of machine learning algorithms into automated data analysis to provide deeper insights and more actionable recommendations.

1.2. Objective

The primary objective of this study was to develop a Chrome extension that integrates session recordings with eye tracking to provide comprehensive user behavior insights, which can be used in both practical applications and academic research. To achieve the primary objective, we outlined the following specific objectives:

1. Create a functional Chrome extension using rrweb for session recording and WebGazer.js for eye tracking.
2. Implement features such as session recording, replay with eye-tracking data overlays, session management, and options for exporting and importing recorded sessions.
3. The usability and effectiveness of the extension were evaluated through a structured user study and detailed feedback analysis. The study quantifies the following:

– **Usability characteristics:** Interface Intuitiveness, Feature Accessibility, Performance, Learning Curve, Satisfaction;

– **Core Features:** Session Recording, Session Replay, Eye-Tracking Data Accuracy, Session Management;

– **Recommendation Likelihood**.

1.3. Approach

To achieve these objectives, we developed the Chrome extension using rrweb for session recording and WebGazer.js for eye tracking. The development process involved architectural modeling using UML diagrams, software engineering techniques for extension implementation, and integration testing to ensure seamless operation of the combined features. We also applied data preprocessing techniques to prevent redundancy and reduce noise in eye-tracking data. The usability and effectiveness of the extension were evaluated by conducting a structured user study with 25 participants aged 18 to 35 years. Participants interacted with the extension in a controlled setting. Then, the participants provided feedback through a detailed questionnaire, which included both Likert-scale questions and open-ended responses. The goal of this study was to evaluate the extension's features, user interface, and overall user experience.

This paper has the following structure: Section 2 reviews work related to web session recording and eye-tracking technologies. Section 3 describes the methods used to develop the extension. This includes information about the extension's high-level architecture, the development framework and libraries, and the algorithms used to process the captured eye-tracking data. Section 4 provides a detailed overview of the user interface, highlighting the main components, and explaining how users interact with the extension. Section 5 details the user evaluation and results. Section 6 discusses the findings and implications for future development. Section 7 summarizes the key contributions and outlines potential directions for future research.

2. Literature Review

In the field of Human-Computer Interaction (HCI), various methods and tools have been explored for capturing user interactions and improving usability. For example, [8] provided a detailed overview of HCI methods and principles, highlighting the importance of usability when designing interactive systems.

Understanding user behavior on the web has been a subject of research for many years. Various tools and methodologies have been developed to capture and analyze user interactions, with the aim of enhancing user experience and optimizing web design.

Some studies have demonstrated that traditional methods of collecting user interaction data often cannot provide sufficient details to fully understand user behavior, and this is where eye tracking can fill the gap [1 - 3]. Eye tracking provides an understanding of cognitive processes and user behavior that cannot be observed using traditional interaction data [9]. Research in cognitive psychology has demonstrated the importance of eye movements when reading and processing information [4], highlighting the value of eye-tracking data when understanding user behavior.

Eye-tracking technology is used in various fields, including psychology, human-computer interaction, marketing, and medical research [10], to study where and how users focus their visual attention. It can also be used as a computer input and interaction method [11]. [7] proposed using eye-tracking technology as a

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

biometric tool for user identification, offering an additional layer of security through unique visual patterns. Eye-tracking data can be used to analyze user behavior, identify usability issues [5, 6], enhance cybersecurity [7], and even recognize emotions, either associated with user interactions – [12] or occurring while consuming media content [13]. In [6], it was found that usability problems are connected to a specific sequence of eye-tracking patterns.

A recent study reviewed the use of eye-tracking methods to evaluate the effectiveness of advertising by analyzing 112 papers published between 1979 and 2019. This study shows that eye-tracking techniques provide detailed, moment-by-moment insights into what people look at in ads, thereby reducing the chance of mistakes caused by relying on memory or opinions. This helps us better understand how different elements such as size, color, and gaze direction affect the effectiveness of ads [14].

This section reviews the existing literature on web session recording, eye-tracking technologies, and their integration. It also highlights the gaps we address in this study.

2.1. Web Session Recording

Web session recording captures user interactions on a website to analyze user behavior and identify usability issues. This includes recording user clicks, mouse movements, scrolls, and form inputs. Several tools and frameworks have been developed for this purpose. RRWeb is a popular library for recording and replaying web sessions. It captures the Document Object Model (DOM) mutations, user interactions, and other relevant events, allowing for detailed replay of user sessions. Due to its efficiency, ease of integration, and open-source nature, RRWeb has been widely adopted by various companies and projects. However, although it provides detailed session recordings, it does not have built-in support for eye tracking, which limits its ability to provide a deeper understanding of user attention and focus [15].

The popular tools are Hotjar and FullStory. They offer advanced features for session recording and user feedback and provide instruments for analyzing user interactions, like heatmaps. These tools are widely used in the industry to improve user experience and identify problems in web design. Despite their advanced features, such tools are often not offered for free and do not provide open-source solutions. These factors limit their accessibility to academic research and smaller development projects.

2.2. Eye-Tracking Technologies

OGAMA (Open Gaze and Mouse Analyzer) is an open-source tool for analyzing eye and mouse movements in slideshow-based experimental setups [16]. It is optimized to work best with fixed images or slideshows; thus, its usage is limited when working with dynamic and interactive web content. Additionally, OGAMA is not cross-platform, which creates difficulties when using it on other operating systems.

GazeParser is another open-source tool designed for accurate eye tracking in controlled experimental environments [17]. It uses Python for advanced data analysis and integrates with libraries like PsychoPy. Despite its advantages, its desktop-based nature makes it challenging to integrate it with web session recording tools to provide a complete and easy-to-use solution that combines session recording and eye tracking.

WebGazer.js is an open-source library that allows eye tracking in web browsers using standard webcams. It uses computer vision and machine learning methods to estimate the point at which the user is looking on the screen (gaze point) [18]. WebGazer.js makes eye-tracking more accessible to regular users by eliminating the need for specialized hardware. However, there are problems with accuracy and calibration.

A recent study validated the use of WebGazer.js for research in early childhood. The study found that although the remote web-based method using WebGazer.js resulted in noisier data and higher attrition rates than traditional in-lab eye-tracking systems, it still provided meaningful insights into goal-directed action predictions [19]. Another study found that WebGazer.js provided accurate and precise gaze measurements to predict task-unrelated thought (TUT) and reading comprehension, demonstrating its potential for scalable and affordable real-time cognitive state detection in online educational settings [20].

There are commercial eye-tracking solutions such as Tobii and EyeLink. They offer high precision and reliable data collection abilities. These systems are widely used in research and industry to study user visual attention. However, their high cost and need for special equipment make them less accessible to regular users and academic researchers.

2.3. Integration of Session Recording and Eye-Tracking

Integrating session recordings using eye tracking can provide a more complete view of user behavior. Despite the potential benefits, there are not many tools that seamlessly combine these technologies.

GazeRecorder is an example of a tool that provides webcam-based eye-tracking and web session recording capabilities [21]. It allows real-time tracking and heatmap generation using standard webcams. This makes it a cost-effective and accessible solution for usability studies and behavioral research. However, integrating GazeRecorder requires access to and modification of the website's codebase. In addition, GazeRecorder is not an open-source tool; thus, its flexibility in terms of customization is limited.

iTrace is another example of a tool that integrates eye-tracking within the Integrated Development Environment (IDE) to analyze user behavior during software engineering tasks [22]. It records both session activities and detailed gaze data on software artifacts to provide a comprehensive understanding of developers' interactions and cognitive processes. However, iTrace is strictly specialized for IDE environments, which limits its use to analyze user behavior on web pages.

2.4. Identified Gaps and Contribution of the Current Study

The review of the existing literature revealed a gap in the availability of accessible tools for recording web sessions using eye-tracking data. Although rrweb and WebGazer.js provide basic capabilities for session recording and eye tracking, there is a need for a user-friendly tool that combines these technologies to provide a better understanding of user behavior. The current study addresses this gap by developing a Chrome extension that integrates rrweb and WebGazer.js to provide a solution to record and replay web sessions with integrated eye-tracking data.

One of the significant advantages of the developed Chrome extension is its ability to record web sessions with eye-tracking data on any website without requiring modifications to the website's codebase.

This study contributes to the field of tools for user behavior analysis by evaluating the proposed extension through a structured user study and provides a roadmap for future improvements in this field.

3. Methodology

Our Chrome extension is designed to improve usability testing by recording user session data. The data include mouse movements, clicks, scrolls, navigations, form inputs, and eye-tracking data. The extension captures real-time eye-tracking data and combines them with user interaction events to provide a detailed understanding of user behavior and visual attention on web pages. This dual-data approach allows researchers and designers to analyze not only the actions taken by the user but also what they were looking at on the webpage at the time of these actions.

The Chrome extension works in a way that does not distract user attention while recording a session; thus, user experience is not affected during data collection.

3.1. Development Framework and Libraries

To develop the Chrome extension, we used modern web technologies and libraries to ensure its reliable and effective implementation.

The main framework used to construct the user interface was React, which was combined with TypeScript. React is a popular JavaScript library for building user interfaces. It was selected for its component-based architecture, which helps create reusable components and maintain code modularity. The use of React facilitates the development of dynamic and responsive UI elements. TypeScript is a statically typed superset of JavaScript. It enhances code quality and maintainability by providing type safety and reducing runtime errors.

The extension uses rrweb to record user sessions. Rrweb is a powerful open-source tool that captures and replays web sessions by recording changes in the DOM (Document Object Model) and user interactions. This library was selected for its ability to generate detailed recordings with minimal performance overhead. Thus makes it a well-suited tool for capturing user sessions without affecting the browsing experience.

The integration of the eye-tracking functionality was implemented using WebGather.js. It is an open-source library designed to track a user's gaze using a standard webcam. WebGazer.js uses machine learning models to predict where the user is looking on the screen, providing real-time data. This library was selected for its easy integration and capability to operate directly within the browser, with no need for specialized hardware.

3.2. Extension Architecture

The extension architecture is modular and includes separate components for the user interface (UI), content script, background script, session recording, and eye-tracking. The following list gives an overview of the extension's key components:

1. **User Interface (UI).** It is built using React and TypeScript and provides an interactive interface for users to control the extension. The UI includes features to start/stop session recording, manage recorded sessions, and replay sessions with eye-tracking data overlays.
2. **Session Recording Module.** The proposed model uses rrweb to record DOM mutations and user events. The module initializes rrweb, controls the start and stop of session recording, and sends the captured data to the content script.
3. **Eye-Tracking Module.** It uses WebGazer.js to capture eye-tracking data. The module is responsible for handling the calibration process, the start and stop of

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

capturing eye-tracking data, and sending them to the content script.

4. **Content Script.** It is injected into web pages and is responsible for initialization and communication with session recording and eye-tracking modules to receive recorded DOM mutations and user events, capture eye-tracking data, and forward these data to the background script.

5. **Background Script.** It manages the extension's core logic and handles communication between the UI and content script. The background script is also responsible for receiving and processing session and eye-tracking data and saving the final session data using the local storage module.

6. **Local Storage Module.** It is responsible for storing all recorded sessions and eye-tracking data locally on the user's device. Under the hood, it uses the browser's native "localStorage" provided by Web Storage API. The local storage module provides an API to manage recorded sessions, export session data to a file, and import session data from a file.

Fig. 1 contains a component diagram illustrating the main components of the extension.

Fig. 2 displays a flowchart that represents the workflow of the extension from its start to the storage of recorded session data.

1. **Initialization**. When the extension is activated, the background script initializes the required components and injects the content script into the current web page.

2. **Session Start**. When the user starts the session recording from the extension's popup, the UI sends a message to the background script to begin recording a new session. The background script then sends a message to the content script to initialize the session recording and eye-tracking modules and begin capturing the data.

3. **Data Capture**. The content script receives data from the session recording and eye-tracking modules and forwards these data to the background script.

4. **Session End**. When the user stops recording the session, the UI sends a stop command to the background script. The background script begins processing the session data before saving the data.

5. **Data Processing and Storage**. The background script processes the received data, applies filtering to the eye-tracking data (including moving average filtering, velocity thresholding, and distance-based thinning), synchronizes the session and filtered eye-tracking data, and stores them for later retrieval and replay.

To provide a clear visualization of the session recording process, we included a UML sequence diagram in Fig. 3. This diagram illustrates the interactions between the main components of the extension.

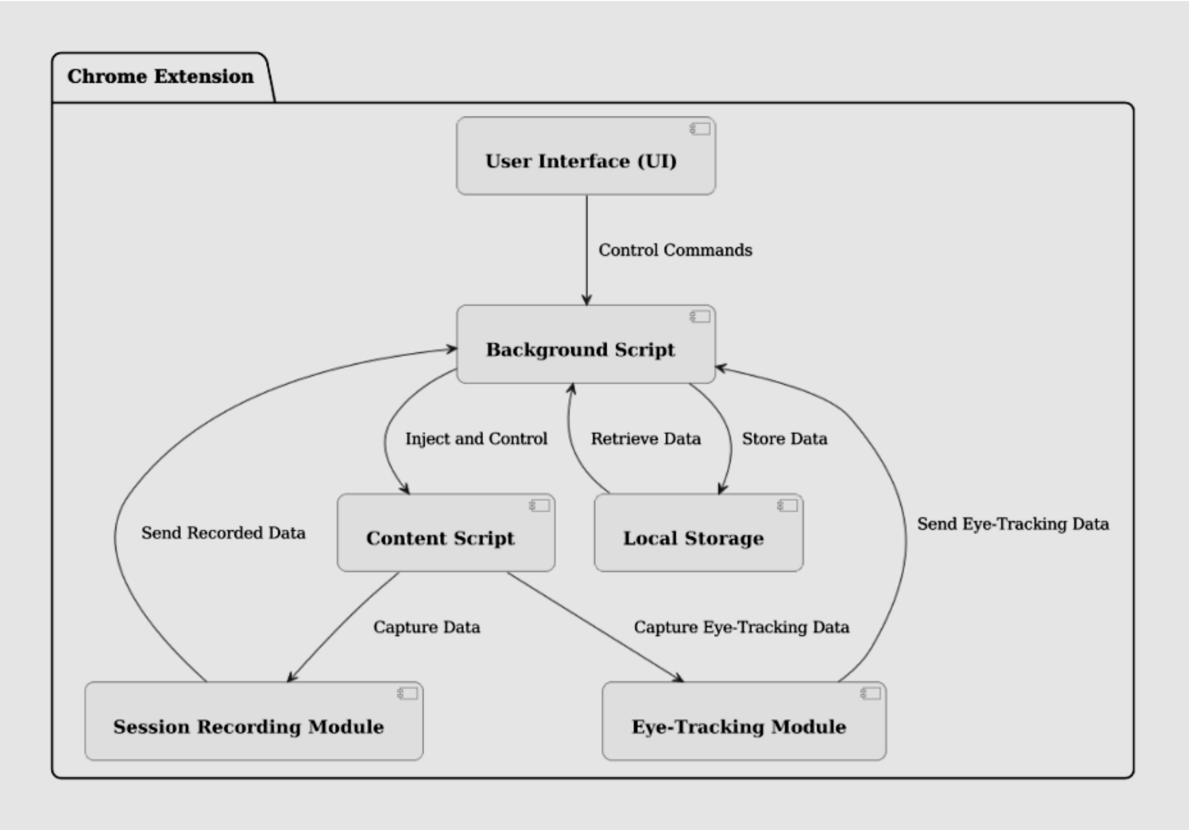

Fig. 1. Component Diagram of the Extension Architecture

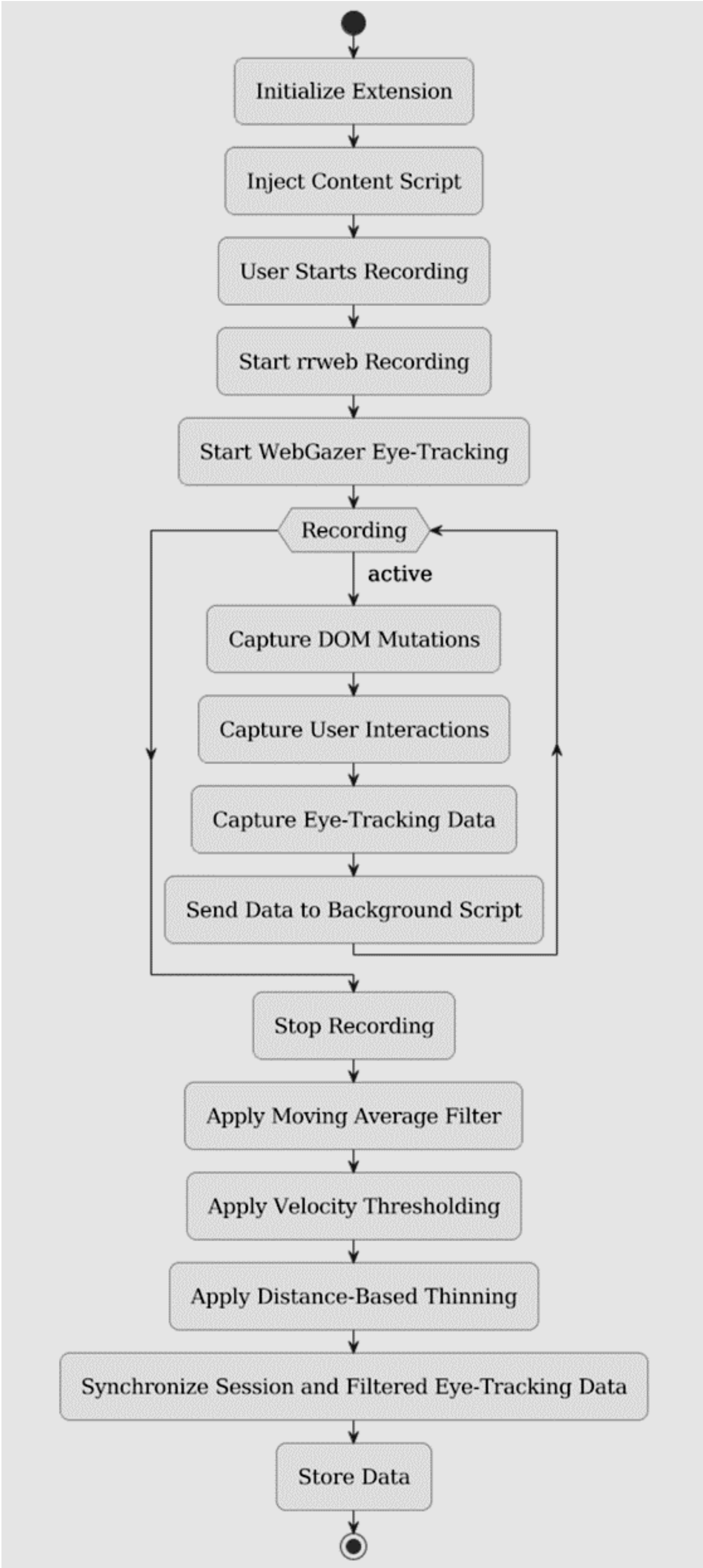

Fig. 2. Flowchart of the Extension's workflow

3.3. Session Recording

For the session recording functionality, the extension uses rrweb, a powerful open-source library designed for recording and replaying web sessions. Rrweb was selected for its ability to accurately record and replay user sessions by capturing DOM mutations, user interactions, and other relevant events. The high

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

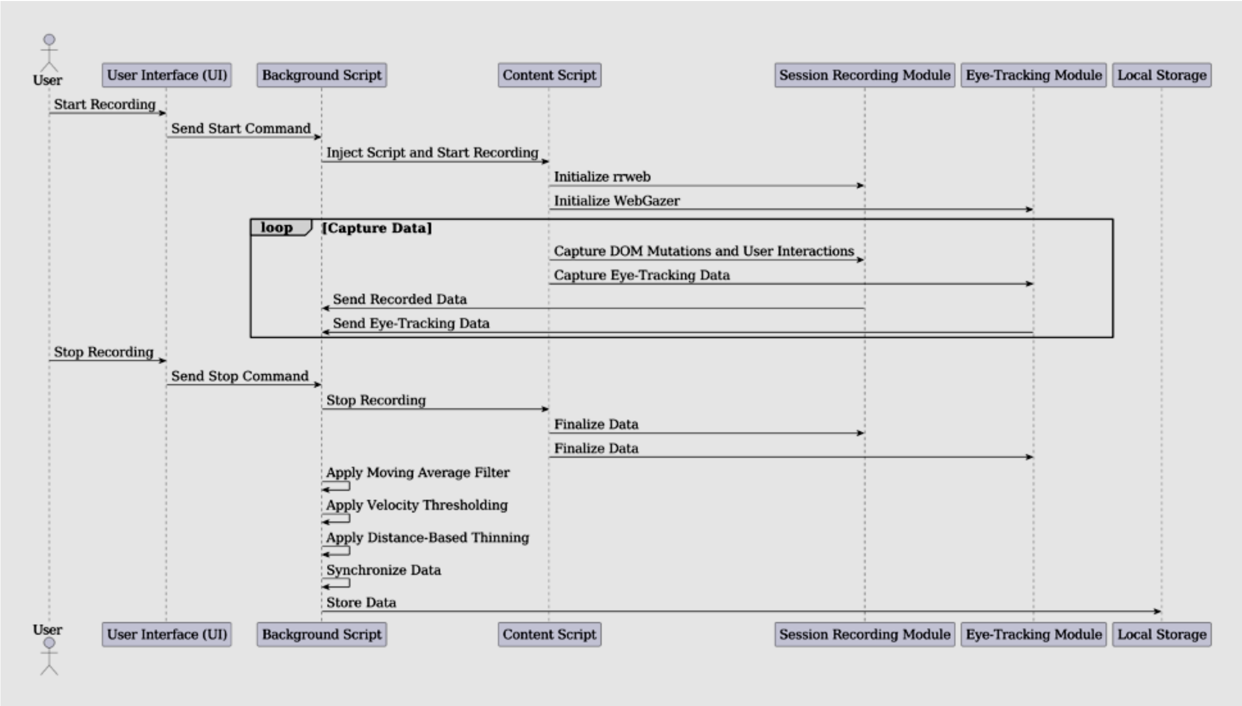

Fig. 3. UML Sequence Diagram of the Session Recording Process

level of fidelity in the recordings ensures that the replayed sessions closely mirror the original user interactions, which is essential when analyzing user behavior. Another key advantage of rrweb is that it is optimized to minimize performance overhead. This ensures that the recording process doesn't negatively impact user browsing experience.

The key features of rrweb include:

1. **DOM Snapshot.** Rrweb periodically takes snapshots of the entire DOM. These snapshots provide a complete view of the web page's structure at specific time intervals.

2. **Incremental DOM Changes.** Between snapshots, rrweb captures incremental changes to the DOM, including modifications, additions, and deletions to elements. This ensures that all dynamic changes to the web page are recorded.

3. **User Interactions.** Rrweb records user interactions, such as clicks, scrolls, inputs, and mouse movements. These interactions are essential for understanding user behavior and actions on the web page.

4. **Custom Events.** In addition to standard interaction events, rrweb allows developers to record and replay user interactions or state changes specific to their applications. Custom events are displayed in the timeline during replay as standard interaction events. Developers, for example, can add custom visualization to these events or use them as debugging information while optimizing user experience.

5. **Event Metadata.** Each recorded event includes metadata, such as timestamps. This allows precise playback synchronization.

3.4. Eye-Tracking Integration

The integration of the eye-tracking functionality was implemented using WebGazer.js. It is an open-source library designed to track a user's gaze using a standard webcam. WebGazer.js uses machine learning models to predict where the user is looking at a screen in real time.

WebGazer.js was selected for this study due to the following key factors:

1. **Cost-Effectiveness**. Unlike specialized eye-tracking hardware, WebGazer.js operates with standard webcams. This capability significantly reduces costs and make the technology more accessible to a broader audience.

2. **Ease of Integration**. As a JavaScript library, WebGazer.js seamlessly integrates with web applications. It provides a straightforward API to start and stop the eye-tracking process and has auxiliary methods for debugging.

3. **Real-Time Tracking**. WebGazer.js provides real-time eye-tracking data. It allows to register rapid changes in the user's gaze and observe them during replay. For web applications that use eye tracking as a means of interaction, real-time data can enhance interactivity and usability.

Despite the advantages of WebGazer.js for use as an eye-tracking tool in the current study, it has the following limitations:

1. **Accuracy Limitations.** Generally, it has lower accuracy than specialized eye-tracking hardware, especially under varying lighting conditions or when using lower-quality webcams.

2. **Calibration Requirement.** The calibration process can be time-consuming and requires user cooperation to obtain reliable eye-tracking results.

The accuracy of eye-tracking data relies heavily on an effective calibration process. WebGazer.js self-calibrates itself when a user interacts with a web page, makes clicks, or performs mouse movements. However, when the user has not yet interacted with the web page, WebGazer.js has no data that can be used to make accurate predictions of the user's gaze. To address this issue, the extension provides a calibration feature to calibrate WebGazer.js prior to recording a session.

During calibration, the web page is covered with a white overlay containing nine red calibration points in different parts of the screen. The user is instructed to focus on each calibration point at a time and perform five clicks. After the user clicks five times on a single calibration point, the point turns green, which indicates that they can move to the next calibration point. WebGazer.js uses each click on a calibration point to train its internal machine-learning model. After the user clicks through the calibration points — the calibration process ends, the calibration overlay closes, and the user can start session recording. Users can rerun calibration by clicking a button in the extension's popup window. Fig. 4 contains the calibration overlay with calibration points. The following items are highlighted numerically:

1. **Not calibrated point.** The user has not yet clicked on it five times. The point is red.

2. **Calibrated point.** The user has clicked on it five times. The point is green.

3. **Gaze prediction point.** The user's gaze point is predicted by WebGazer.js based on the data added during calibration.

By integrating WebGazer.js, the extension uses a practical and accessible eye-tracking solution while acknowledging limitations related to user conditions and calibration process.

3.5. Data Handling and Storage

The data pipeline comprises the following steps:

1. **Capture**. During session recording, session and eye-tracking data are captured by relevant modules. The session data included DOM mutations and user interactions.

2. **Transmission**. The captured data are transmitted in real time to the background script for processing. The proposed method allows continuous data collection without overloading the client's memory.

3. **Processing**. The background script processes the eye-tracking data by applying filtering and smoothing algorithms and synchronizes the session data with the processed eye-tracking data.

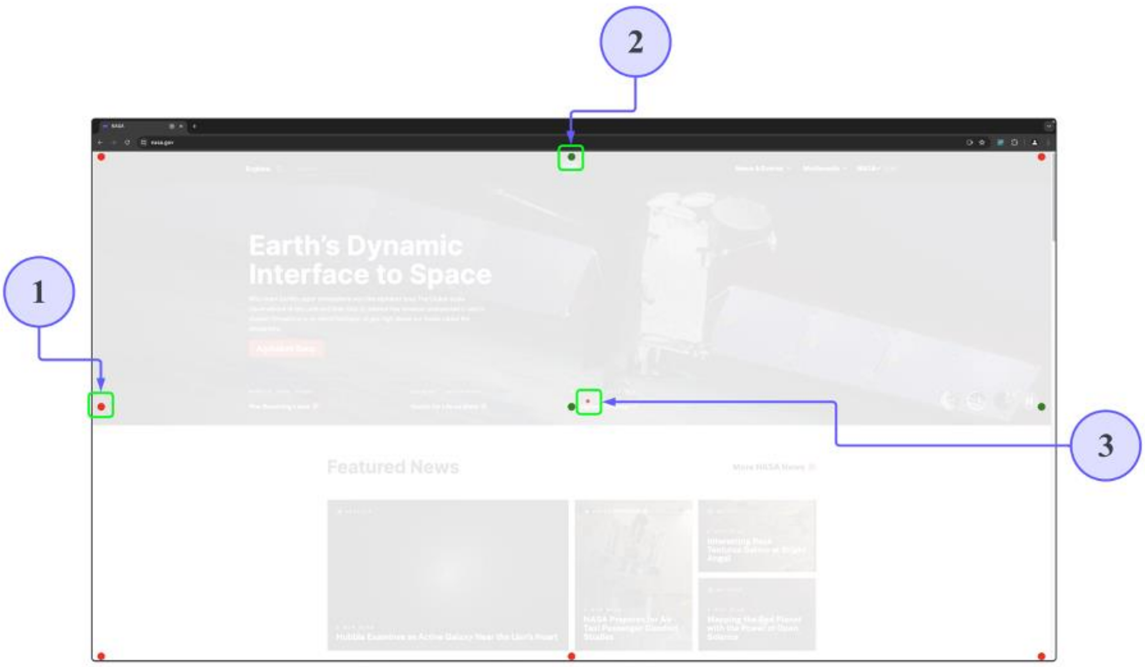

Fig. 4. Eye-Tracking Calibration Overlay

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

4. **Storage**. The processed data are serialized and stored locally on the user's device in a predefined format.

All session and eye-tracking data are stored locally on the user's device. This approach allows quick data access and manipulation without relying on network connectivity. To enhance data portability and user control, the extension provides features for exporting and importing recorded sessions in files. Users can export their session data to a file for backup or analysis or can import recorded sessions from a file.

To prevent data redundancy and reduce noise in eye-tracking data, the following filters were applied during the processing step:

1. **Moving Average Filter**. This filter was applied to smooth out short-term variations in the eye-tracking data. The average position of the gaze point is calculated in a sliding window of size five frames. It provides a more stable and continuous gaze path. The window size was chosen experimentally to balance the smoothing efficiency and data responsiveness.

2. **Velocity Thresholding**. This filtering technique removes gaze points where the eye movement velocity is below a threshold of 15 pixels per frame. Removing these minor movements made the data more focused on significant eye movements, increasing the analysis's reliability. The threshold was empirically determined to remove small involuntary movements while maintaining meaningful eye movements.

3. **Distance-Based Thinning**. This filter only retains gaze points that are at least 20 pixels apart from the last retained point. This reduces data volume while preserving significant eye movement patterns. The distance threshold was chosen experimentally to balance data reduction and keep eye movements, which are essential for analysis.

Fig. 5 illustrates the sequence of filters applied to the raw eye-tracking data.

3.6. Data Privacy and Security Measures

It is difficult to overstate the importance of privacy and security when handling user data. Since the extension collects session and eye-tracking data, reliable measures to protect users' personal information are essential. The extension uses a few techniques to minimize the amount of sensitive data.

Minimizing Data Collection. The extension is designed to collect only the data required for analysis. It avoids capturing unnecessary personal information and focuses exclusively on user interactions and gaze data.

Data Anonymization. To ensure that individual users cannot be identified from the data, the following measures are used for session and eye-tracking data:

– **Session Data.** By default, rrweb masks sensitive input fields and excludes them from the recorded session data. Sensitive input fields contain names, email addresses, and other personally identifiable information (PII). For example, when a user enters their email address or password in a form, these fields are masked, and the entered values are not recorded. Only non-identifiable session data, such as mouse clicks and movements, page scrolls, and DOM mutations, are recorded;

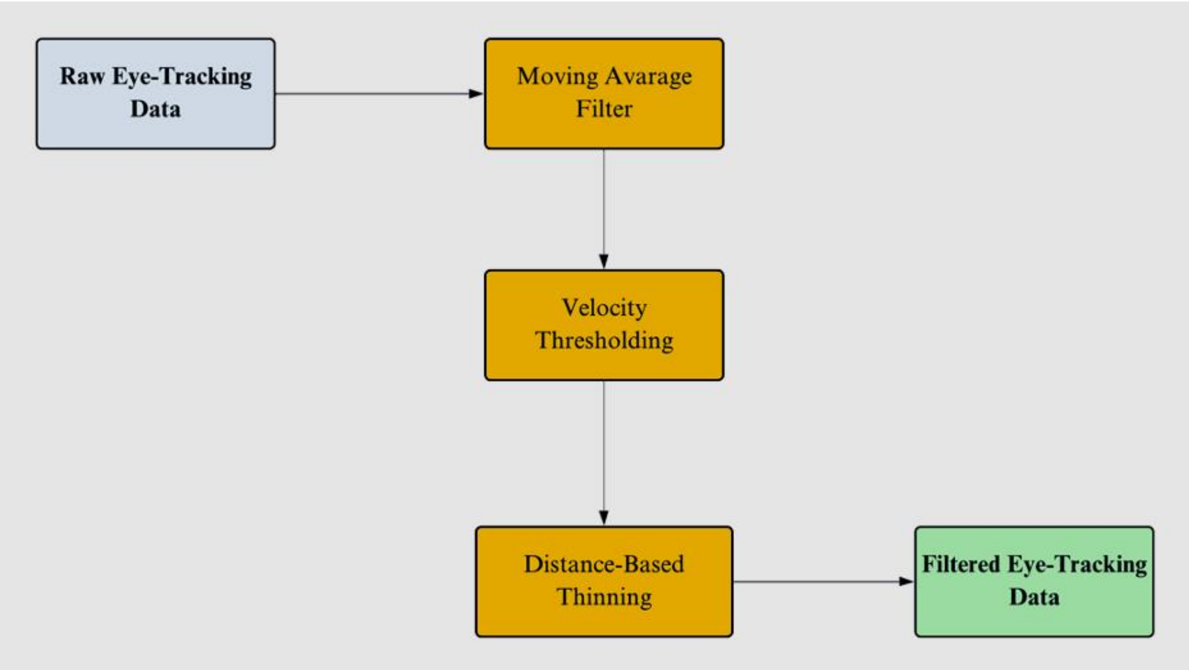

Fig. 5. Gaze Data Filtering Pipeline

– **Gaze Data.** Eye-tracking data collected via WebGazer.js contains only gaze coordinates. No user images or videos are captured. All eye-tracking data are processed locally on the user's device, and only gaze coordinates are stored, which guarantees that the data remain anonymous.

4. User Interface

The user interface (UI) of the developed extension is designed to be intuitive and user-friendly. It consists of several components that allow users to control the recording process, manage recorded sessions and replay them. This section provides an overview of these components and highlights key features with screenshots of the developed extension's UI.

4.1. Controlling the Recording Process

This extension provides a user interface to control the session recording process using the start and stop buttons. The UI also displays the session recording status.

Fig. 6 displays a screenshot of the browser window with the extension popup window open. The following elements are numerically highlighted:

1. **Extension popup window.**
2. **Start/Stop button.** Controls the recording status of a session.
3. **Session recording status.** The pulsating red circle is displayed when the session is recorded.
4. **Session duration.** The duration of recording session is displayed.
5. **"Record session" navigation tab.** Navigates users to this view.
6. **"Recorded sessions" navigation tab.** Navigates users to the list of recorded sessions. It also displays a badge with the number of recorded sessions locally stored on the user's device.
7. **"Calibrate" button.** Starts the eye-tracking calibration process.

4.2. Managing Recorded Sessions

The proposed extension provides a user interface to manage recorded sessions. It contains a list of recorded sessions, from which users can export, delete, or import a session from a file.

Fig. 7 displays a screenshot of a browser window with the extension popup open, displaying the user interface for managing recorded sessions. The following elements are numerically highlighted:

1. **Recorded session item.** It displays information about the recorded session, such as the webpage title, webpage URL, date and time of recording, and duration.

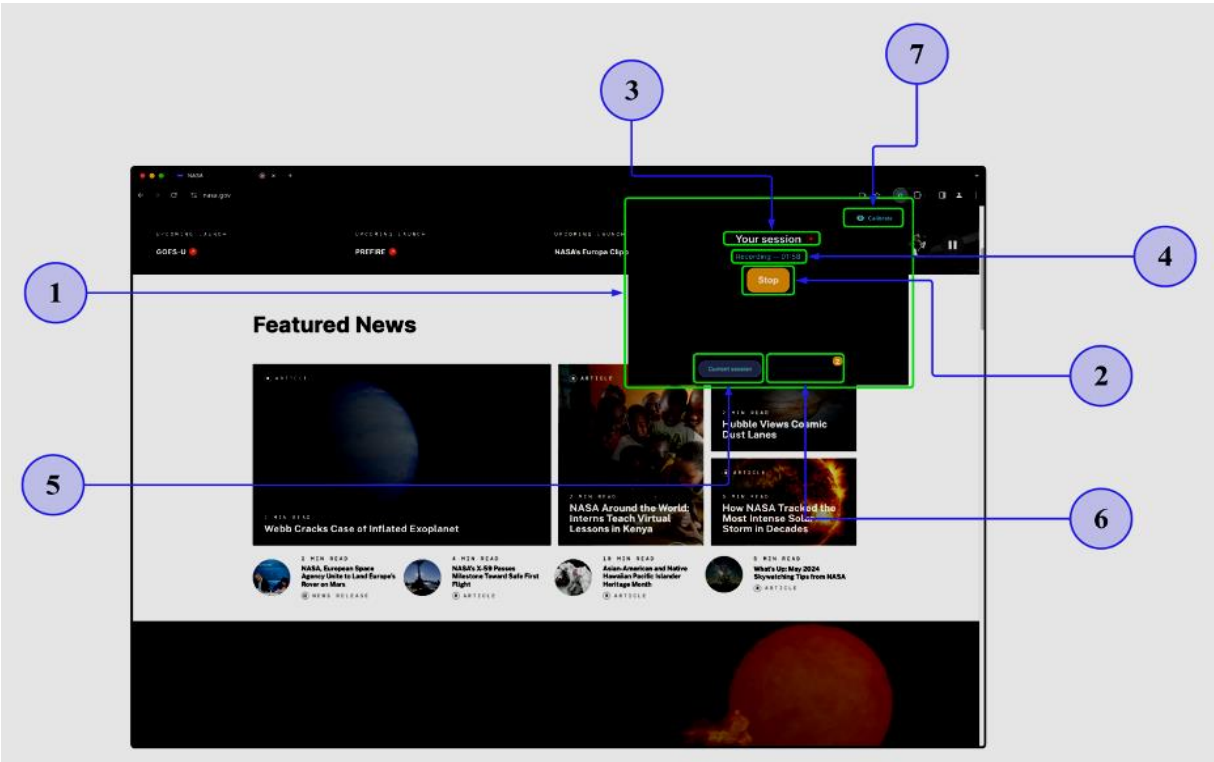

Fig. 6. Extension Popup Window - Recording Session Interface

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

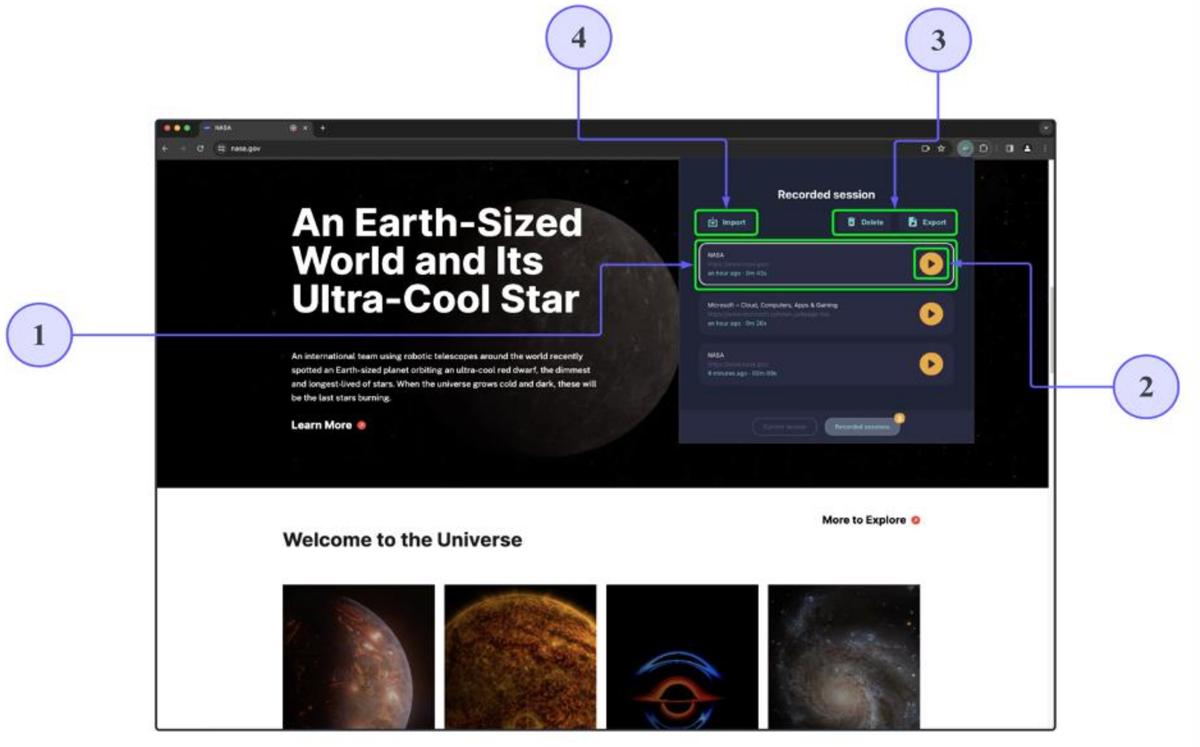

Fig. 7. Extension Popup Window - Session Management Interface

1. **Session replay button.** Once clicked, a new browser tab is opened with the Session Replay Tool, which allows users to replay the selected session.

2. **Controls to export and delete selected sessions.** By clicking the "Export" button, users can export the selected session as a JSON file. The "Delete" button permanently deletes the recorded session from the user's device.

3. **"Import session" button.** Recorded sessions can be imported from a JSON file. This feature allows users to import sessions recorded on other devices.

4.3. Session Replay Tool

The extension provides a user interface to replay recorded sessions. Once the "Session replay button" is clicked, the extension opens a new browser tab with the session replay tool, from which users can replay the selected session.

Fig. 8 displays a screenshot of the browser tab with the session replay tool. The following elements are numerically highlighted:

1. **Mouse pointer indicator.** Displays the position of user mouse pointer during the session.
2. **Eye-gaze point indicator.** Displays the user's gaze point during the session.
3. **Timeline slider**. Allows users to navigate the recorded session.
4. **Other controls**. They include controls for playing/pausing the session replay, controlling the playback speed, and skipping inactivity intervals.

5. Evaluation and Results

To evaluate the usability and effectiveness of the developed extension, a structured user study was conducted. Participants interacted with the extension in a controlled setting. After that, they completed a detailed questionnaire.

5.1. Methodology

Participants were selected to represent potential end-users of the extension, having different levels of expertise and familiarity with technology. A total of 25 participants aged 18–35 years were recruited for the study. The participants included both novice and experienced users to ensure a diverse range of feedback.

Each participant was asked to perform specific actions using the extension, including:

– **Recording a session.** Users went through an eye-tracking calibration process, started the recording process, interacted with a web page, and then stopped the recording;

– **Replaying a session.** The user replayed the recorded session to review the captured interactions and eye-tracking data;

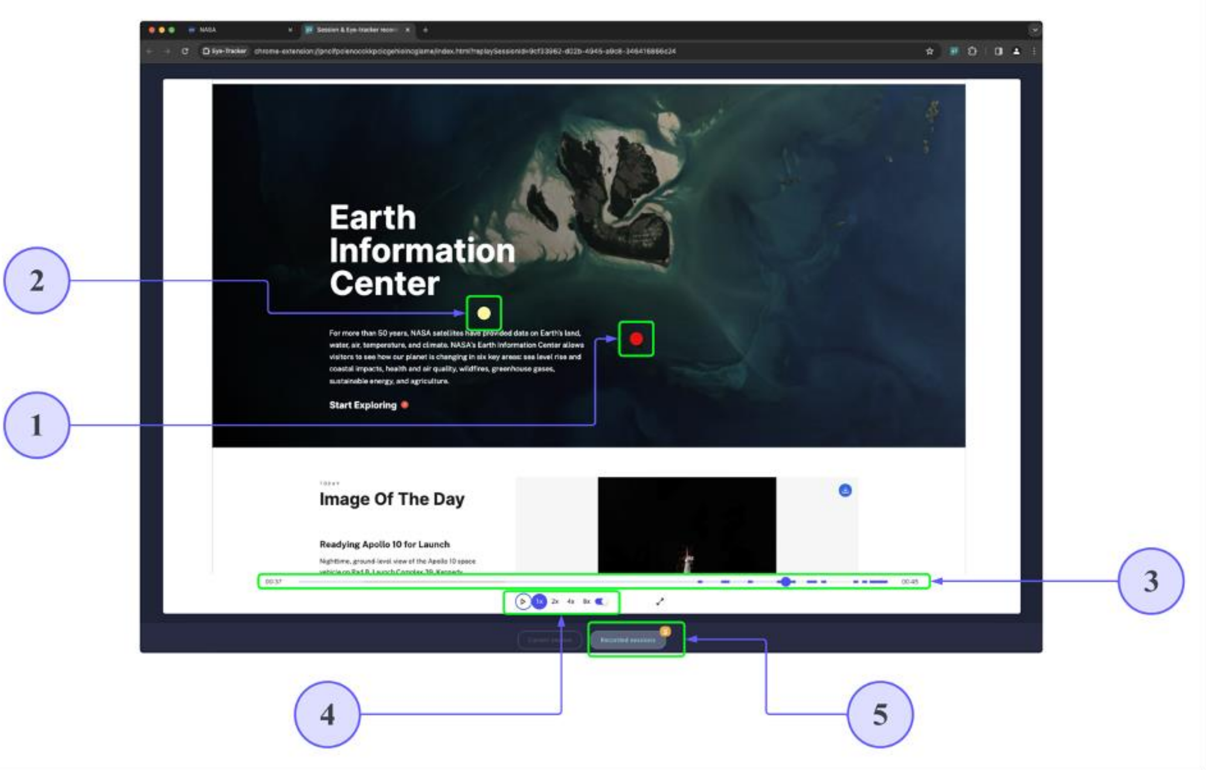

Fig. 8. Session Replay Tool - Replaying Recorded Session Interface

– **Managing sessions.** Users were asked to export, import, and delete recorded sessions to evaluate the session management feature.

5.2. Data Collection

The user feedback was collected using a questionnaire, including Likert-scale questions and open-ended responses. This approach helped in conducting both quantitative and qualitative analyses.

Participants rated their experiences on a Likert scale ranging from 1 to 5, with the option to use decimal values for more precise feedback. The following list includes the characteristics that were evaluated:

– **Interface Intuitiveness.** How easy it is for users to understand and navigate the interface.

– **Feature Accessibility.** How easily users can access and use the features of the extension.

– **Performance.** The responsiveness and efficiency of the extension are during use.

– **Learning Curve.** How quickly users can learn to use the extension effectively.

– **Satisfaction.** Users' overall satisfaction with the extension.

In addition to these overall usability characteristics, participants were asked to rate specific core features of the extension, including the following:

– **Session Recording.** The effectiveness of the session recording functionality.

– **Session Replay.** The effectiveness of the session replay functionality.

– **Eye-Tracking Data Accuracy.** Accuracy of the captured eye-tracking data.

– **Session Management.** The ease of managing sessions, including viewing, deleting, importing, and exporting recorded sessions.

Participants were also asked about their likelihood of recommending the extension to others. This metric helps assess overall user satisfaction and acceptance.

In the open-ended responses section of the questionnaire, participants provided more detailed feedback about their ratings and issues with the extension they faced during the study and were able to suggest improvements.

5.3. Results

The average user scores for each usability characteristic are presented in Table 1.

The average user scores for each core feature are presented in Table 2.

Analysis of the quantitative data revealed that the majority of users found the extension user-friendly (Fig. 9). "User-friendly" here refers to the overall ease of use and positive user experience, as reflected in the average usability rating. The usability rating is the average of user scores for the aforementioned criteria measured on a Likert scale (1 to 5). The extension received an average usability rating of 4.5 out of 5.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Table 1
Average User Scores for Usability Characteristics

Usability Characteristic	Average Score
Interface Intuitiveness	4.4
Feature Accessibility	4.4
Performance	4.5
Learning Curve	4.5
Satisfaction	4.5

Table 2
Average User Scores for Core Features

Core Feature	Average Score
Session Recording	4.1
Session Replay	4.3
Eye-Tracking Data Accuracy	3.8
Session Management	4.2

The analysis of feature-specific ratings revealed that most features were well received (Fig. 10).

The specific features of the extension received the following ratings:

– **Session Recording:** 4.1 out of 5.
– **Session Replay:** 4.3 out of 5.
– **Eye-Tracking Data Accuracy:** 3.8 out of 5.
– **Session Management:** 4.2 out of 5.

Qualitative feedback underscored the usefulness of the eye-tracking feature; however, some users noted inaccuracies in the data, which is reflected in a lower rating for this feature. The most common responses were suggestions for improvements, particularly regarding the accuracy of the eye-tracking data.

The likelihood of users recommending the extension to others was high, which indicates overall user satisfaction and acceptance (Fig. 11).

6. Discussion

The user study conducted to evaluate the Chrome extension for recording and replaying web sessions using eye-tracking data integration provided valuable insights into its usability and effectiveness. Overall, participants reported that the extension was effective, and most features received high ratings. However, the results of this study helped identify areas for improvement, particularly problems with occasional inaccuracies in eye-tracking data.

6.1. Usability and User Experience

The proposed extension received high ratings for interface intuitiveness, feature accessibility, performance, learning curve, and overall satisfaction. The average usability rating (of 4.5 out of 5) indicates that participants generally found the extension easy to use and effective in performing its functions. The high ratings for session recording (4.1), session replay (4.3), and session management (4.2) further underscore the positive user experience.

Fig. 9. Bar Chart of Overall Usability Ratings

Fig. 10. Box Plot of Feature-Specific Ratings

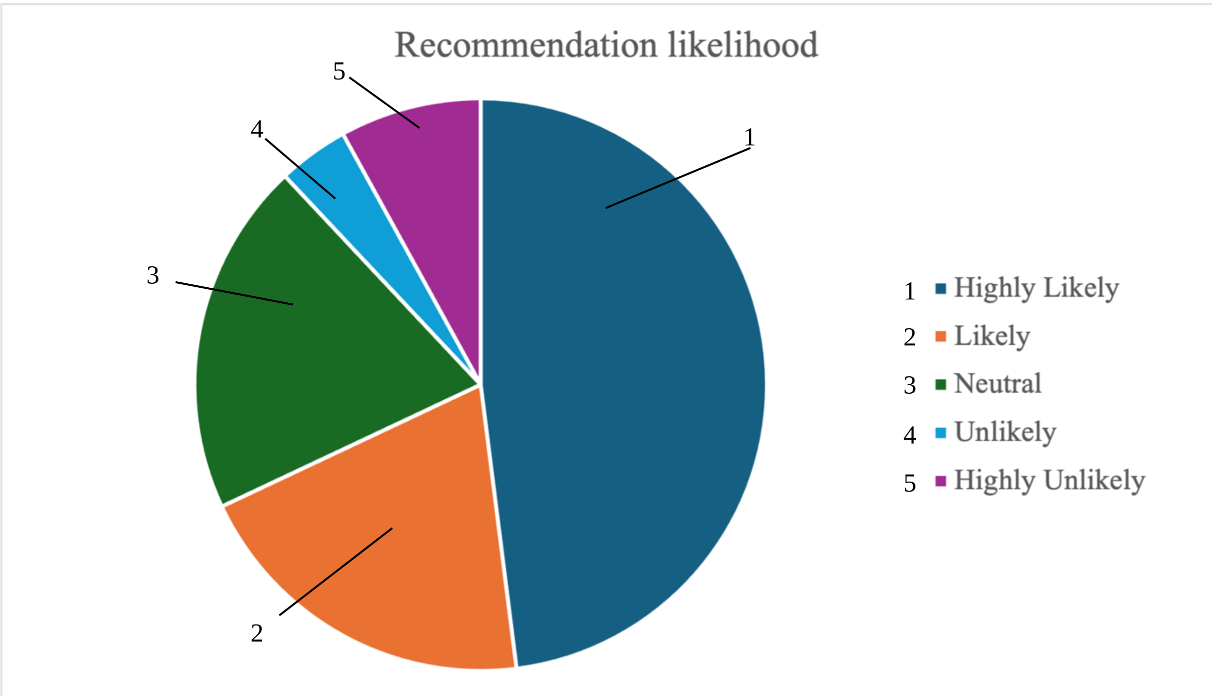

Fig. 11. Pie Chart of Recommendation Likelihood

6.2. Challenges in Eye-Tracking Accuracy

Despite the overall positive feedback, the eye-tracking feature received a lower average rating of 3.8. Some users reported inaccuracies in their eye-tracking data, which occasionally did not align well with their actual gaze points. This issue highlights an important area for improvement because accurate eye-tracking data are crucial for the extension to provide detailed information about user interactions.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

6.3. Implications and Recommendations for Future Development

Based on the analysis of user feedback presented in the results section, several important areas for improvement were identified. Improvements in these areas are necessary to enhance the extension's overall usability and effectiveness:

1. **Enhanced Calibration for Eye-Tracking.** Implement a more reliable calibration process with additional calibration points. This improves the accuracy of eye-tracking data for each user during their session. Improved calibration reduces errors and ensures more precise gaze tracking, which provides higher-quality eye-tracking data for analysis.

2. **Enhanced Data Visualization.** Integration of additional visualization tools, such as heatmaps and gaze plots, to provide a deeper understanding of user behavior. In addition, using machine learning models can help automatically identify user behavior patterns and areas of interest, which will improve the analysis of user sessions and provide more useful insights.

7. Conclusion

This paper presented the development, implementation, and evaluation of a Chrome extension designed to record and replay web sessions using integrated eye-tracking data. The primary objective was to develop a tool that integrates session recordings with eye tracking to provide comprehensive user behavior insights, which can be used in practical applications and academic research.

The extension was constructed using React with TypeScript, and it utilized libraries such as rrweb for session recording and WebGazer.js for eye tracking. The key features of the extension include session recording, session replay with eye-tracking overlays, and session management with the ability to import and export sessions.

A structured user study with 25 participants aged 18 to 35 years was conducted to evaluate the usability and effectiveness of the extension. The extension received high usability ratings, with an average score of 4,5 out of 5, indicating that users found the extension user-friendly and effective. Specific features such as session recording, session replay, and session management, were well received, whereas the eye-tracking feature received a lower rating of 3.8 due to occasional inaccuracies in the eye-tracking data.

In summary, the Chrome extension developed in this study is a valuable tool for recording and replaying web sessions with integrated eye-tracking data. Positive user feedback and high usability ratings confirm the effectiveness of the proposed model in analyzing user behavior on the web. However, the study also identified areas for improvement, particularly in terms of the accuracy of the eye-tracking feature. Future enhancements based on user feedback will focus on improving eye-tracking accuracy and integrating machine learning algorithms to automatically analyze recorded data, providing deeper insights and actionable recommendations. These improvements will enhance the extension, making it a more effective tool for analyzing user behavior on the web.

Contributions of authors: conceptualization, methodology – **Serhii Zelinskyi**, **Yuriy Boyko**; formulation of tasks, analysis – **Serhii Zelinskyi**, **Yuriy Boyko**; development of the Chrome extension, coding, testing – **Serhii Zelinskyi**; analysis of results, visualization – **Serhii Zelinskyi**; writing – original draft preparation, writing – review and editing – **Serhii Zelinskyi**, **Yuriy Boyko**.

Conflict of Interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

This study was conducted without financial support.

Data Availability

The work has associated data in the data repository.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence methods while creating the presented work.

All the authors have read and agreed to the published version of this manuscript.

References

1. Schiessl, M., Duda, S., Thölke, A., & Fischer, R. Eye tracking and its application in usability and media research. *MMI Interakt.*, 2003, vol. 6. Available at: https://api.semanticscholar.org/CorpusID:501543. (accessed Mar. 08, 2024).

2. Pretorius, M., Calitz, A. P., & van Greunen, D. *The added value of eye tracking in the usability evaluation of a network management tool*. 2005. Available at: https://api.semanticscholar.org/CorpusID:9747314. (accessed Mar. 08, 2024).

3. Murali, S., Walber, T., Schaefer, C., & Lim, S. H. Enriching Verbal Feedback from Usability Testing: Automatic Linking of Thinking-Aloud Recordings and Stimulus using Eye Tracking and Mouse Data. *ArXiv*, 2023, vol. abs/2307.05171. Available at: https://api.semanticscholar.org/CorpusID:259766058. (accessed Mar. 08, 2024).

4. Rayner, K. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 1998, vol. 124, iss. 3, pp. 372–422. DOI: 10.1037/0033-2909.124.3.372.

5. Wang, J., Antonenko, P., Celepkolu, M., Jimenez, Y., Fieldman, E., & Fieldman, A. Exploring Relationships Between Eye Tracking and Traditional Usability Testing Data. *J Hum Comput Interact*, 2019, vol. 35, no. 6, pp. 483–494. DOI: 10.1080/10447318.2018.1464776.

6. Ehmke, C., & Wilson, S. M. Identifying web usability problems from eye-tracking data. *British Computer Society Conference on Human-Computer Interaction*, 2007. Available at: https://api.semanticscholar.org/CorpusID:1427673. (accessed Mar. 08, 2024).

7. Gordieiev, O., Kharchenko, V., Illiashenko, O., Morozova, O., & Gasanov, M. Concept of Using Eye Tracking Technology to Assess and Ensure Cybersecurity, Functional Safety and Usability. *International Journal of Safety and Security Engineering*, 2021, vol. 11, no. 4, pp. 361-367. DOI: 10.18280/ijsse.110409.

8. Dix, A., Finlay, J., Abowd, G. D., & Beale, R. *Human-Computer Interaction*, 3rd ed. Pearson Education Publ., 2004. 834 p. Available at: https://paragnachaliya.in/wp-content/uploads/2017/08/HCI_Alan_Dix.pdf. (accessed Mar. 08, 2024).

9. Poole, A., & Ball, L. J. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In book: *Encyclopedia of Human-Computer Interaction*, Idea Group Reference Publ., 2005, pp. 211-219.

10. Punde, P. A., Jadhav, M. E., & Manza, R. R. A study of eye tracking technology and its applications. *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, IEEE, Oct. 2017, pp. 86–90. DOI: 10.1109/ICISIM.2017.8122153.

11. Cantoni, V., & Porta, M. Eye tracking as a computer input and interaction method. *15th International Conference on Computer Systems and Technologies*, New York, NY, USA: ACM, Jun. 2014, pp. 1-12. DOI: 10.1145/2659532.2659592.

12. Novák, J. Š., Masner, J., Benda, P., Šimek, P., & Merunka, V. Eye Tracking, Usability, and User Experience: A Systematic Review. *J Hum Comput Interact*, 2023. DOI: 10.1080/10447318.2023.2221600.

13. Tarnowski, P., Kołodziej, M., Majkowski, A., & Rak, R. J. Eye-Tracking Analysis for Emotion Recognition. *Comput Intell Neurosci*, 2020, vol. 2020, pp. 1-13. DOI: 10.1155/2020/2909267.

14. Casado-Aranda, L.-A., Sánchez-Fernández, J., & Ibáñez-Zapata, J.-Á. Evaluating Communication Effectiveness Through Eye Tracking: Benefits, State of the Art, and Unresolved Questions. *International Journal of Business Communication*, 2023, vol. 60, iss. 1, pp. 24-61. DOI: 10.1177/2329488419893746.

15. *"rrweb"*. Github. Available at: https://github.com/rrweb-io/rrweb. (accessed Mar. 08, 2024).

16. Voßkühler, A., Nordmeier, V., Kuchinke, L., & Jacobs, A. M. OGAMA (Open Gaze and Mouse Analyzer): Open-source software designed to analyze eye and mouse movements in slideshow study designs. *Behav Res Methods*, 2008, vol. 40, iss. 4, pp. 1150-1162. DOI: 10.3758/BRM.40.4.1150.

17. Sogo, H. GazeParser: an open-source and multiplatform library for low-cost eye tracking and analysis. *Behav Res Methods*, 2013, vol. 45, iss. 3, pp. 684-695. DOI: 10.3758/s13428-012-0286-x.

18. Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. WebGazer: Scalable Webcam Eye Tracking Using User Interactions. *25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 3839-3845. Available at: https://www.ijcai.org/Proceedings/16/Papers/540.pdf. (accessed Mar. 08, 2024).

19. Steffan, A., Zimmer, L., Arias-Trejo, N., & et al. Validation of an open source, remote web-based eye-tracking method (WebGazer) for research in early childhood. *Infancy*, 2024, vol. 29, iss. 1, pp. 31-55. DOI: 10.1111/infa.12564.

20. Hutt, S., Wong, A., Papoutsaki, A., Baker, R. S., Gold, J. I., & Mills, C. Webcam-based eye tracking to detect mind wandering and comprehension errors. *Behav Res Methods*, 2024, vol. 56, no. 1, pp. 1-17. DOI: 10.3758/s13428-022-02040-x.

21. *Online Eye Tracking Software*. GazeRecorder. Available at: https://gazerecorder.com. (accessed Mar. 08, 2024).

22. Shaffer, T. R., Wise, J. L., Walters, B. M., Müller, S. C., Falcone, M., & Sharif, B. iTrace: enabling eye tracking on software artifacts within the IDE to support software engineering tasks. *2015 10th Joint Meeting on Foundations of Software Engineering*, New York, NY, USA, ACM, Aug. 2015, pp. 954-957. DOI: 10.1145/2786805.2803188.

Received 09.07.2024, Accepted 20.08.2024

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

ІНТЕГРАЦІЯ ЗАПИСУ СЕАНСУ ТА ВІДСТЕЖЕННЯ ПОГЛЯДУ: РОЗРОБКА ТА ОЦІНКА РОЗШИРЕННЯ ДЛЯ CHROME ДЛЯ АНАЛІЗУ ПОВЕДІНКИ КОРИСТУВАЧІВ

С. В. Зелінський, Ю. В. Бойко

Предметом статті є розробка, впровадження та оцінка розширення для Chrome, призначеного для запису та відтворення веб-сесій з інтегрованими даними відстеження погляду. **Метою** є створення інструменту, що покращує аналіз взаємодії користувачів шляхом поєднання запису сесій із можливостями відстеження погляду. **Завдання**, які потрібно вирішити, такі: створити функціональне розширення для Chrome, що використовує rrweb для запису сесій та WebGazer.js для відстеження погляду; впровадити функції запису сесій, відтворення з накладанням даних відстеження погляду та управління сесіями; надати можливості експорту та імпорту записаних сесій. Використовуваними **методами** є: архітектурне моделювання з використанням UML-діаграм для проектування системної архітектури, техніки програмної інженерії для розробки розширення, інтеграційне тестування для забезпечення безперебійної роботи об'єднаних функцій та методи попередньої обробки даних для запобігання надмірності та зменшення шуму в даних відстеження погляду. Крім того, для оцінки зручності використання та збору відгуків було проведено структуроване користувацьке дослідження з детальними анкетами, що поєднують питання за шкалою Лайкерта та відкриті відповіді, а також аналіз відгуків. Отримано такі **результати**: розширення було успішно розроблено та оцінено за участю 25 учасників віком від 18 до 35 років у контрольованому середовищі. Було отримано високі оцінки зручності використання, із середнім балом 4,5 з 5 для функцій запису сесій, відтворення та управління сесіями. Однак функція відстеження погляду отримала нижчу оцінку – 3,8 з 5 через періодичні неточності у даних відстеження погляду. Якісний зворотний зв'язок вказав на корисність функції відстеження погляду, одночасно підкреслюючи необхідність підвищення точності даних. **Висновки.** Наукова новизна цієї роботи полягає в інтеграції запису сесій та відстеження погляду в межах Chrome розширення, що пропонує новий і всеохоплюючий інструмент для аналізу взаємодії користувачів у вебсередовищі. Здатність інструменту захоплювати як поведінкові дані, так і візуальну увагу без необхідності модифікації коду вебсайтів робить його особливо цінним для дослідників, маркетологів, дизайнерів інтерфейсів та розробників продуктів. Дослідження зручності використання та аналіз відгуків надали чіткий напрямок для майбутніх покращень, включаючи підвищення точності відстеження погляду, інтеграцію розширених аналітичних можливостей та налаштовуваних параметрів звітності. **Майбутня робота** також буде спрямована на інтеграцію алгоритмів машинного навчання для автоматичного аналізу записаних даних з метою глибшого розуміння поведінки користувача та надання практичних рекомендацій.

Ключові слова: запис веб-сесій; відстеження погляду; відтворення веб-сесій; аналіз поведінки; тестування зручності використання.

Зелінський Сергій Вячеславович – асп. каф. комп'ютерної інженерії факультету радіофізики, електроніки та комп'ютерних систем, Київський національний університет імені Тараса Шевченка, Київ, Україна.

Бойко Юрій Володимирович – канд. фіз.-мат. наук, доц., зав. каф. комп'ютерної інженерії факультету радіофізики, електроніки та комп'ютерних систем, Київський національний університет імені Тараса Шевченка, Київ, Україна.

Serhii Zelinskyi – PhD Student of the Department of Computer Engineering, Faculty of Radiophysics, Electronics and Computer Systems, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: sv.zelinskyi@gmail.com, ORCID: 0009-0003-6271-2601.

Yuriy Boyko – Candidate of Physical and Mathematical Sciences, Associate Professor, Head of the Department of Computer Engineering, Faculty of Radiophysics, Electronics and Computer Systems, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: boyko@univ.kiev.ua, ORCID: 0000-0003-1417-7424, Scopus Author ID: 24722552300.