Gaze Insights in XR: Real-Time Eye-Tracking Analytics with Elasticsearch

Conference Paper · March 2025

8 authors, including:

Mohammad Jahed Murad Sunny
University of Arkansas at Little Rock

Jayasri Sai Nikitha Guthula
University of Arkansas at Little Rock

Atit Kharel
University of Arkansas at Little Rock

Meherun Nesa Shraboni
University of Arkansas at Little Rock

Gaze Insights in XR: Real-Time Eye-Tracking Analytics with Elasticsearch

M. J. M. Sunny* Jayasri Sai Nikitha Guthula† Atit Kharel‡ Meherun Nesa Shraboni§ Hadi Rashid¶
Praveshika Bhandari‖ Jan P. Springer** Aryabrata Basu††

Emerging Analytics Center, UA Little Rock

Abstract

Gaze tracking serves as a critical component in human-computer interaction, significantly enhancing the capabilities of immersive extended-reality (XR) applications. Despite the integration of advanced eye-tracking technologies in modern XR devices, challenges surrounding usability, scalability, and privacy remain unresolved. This study presents a technical framework leveraging the Elastic Stack (Elasticsearch, Logstash, Kibana) to enable real-time data collection, indexing, and analysis of gaze-tracking data alongside other interaction metrics. The framework integrates scalable distributed storage, efficient querying, and immersive heatmap visualizations within Unity, offering precise insights into user gaze behavior during interactive tasks. A pilot study involving a gaze-intensive VR task validated the system's capacity to identify focus regions, analyze attention shifts, and detect performance bottlenecks. Notable functionalities include real-time heatmap overlays for user engagement analysis and replay systems for assessing spatial navigation strategies. The framework employs secure indexing configurations to address ethical concerns and discusses privacy-preserving mechanisms, including homomorphic encryption, ensuring the protection of sensitive data such as user focus and intent. This pilot provides a robust, scalable, and privacy-compliant solution for gaze-driven analytics, advancing the state of usability studies and XR application development.

Index Terms: Gaze tracking, Extended reality (XR), Human-computer interaction, ELK Stack, Privacy-preserving technologies, Homomorphic encryption

1 Introduction

Extended reality (XR) technologies are revolutionizing human-computer interaction (HCI) by enabling immersive experiences in entertainment, education, healthcare, and enterprise collaboration. Among these advancements, gaze tracking stands out as a transformative tool, enhancing system responsiveness, delivering personalized content, and improving user engagement [37, 39]. However, managing the large volumes of data generated by gaze tracking, while ensuring privacy protections, remains a critical challenge.

Modern devices like the Meta Quest Pro [9], equipped with advanced gaze-tracking sensors, enable real-time gaze-based interactions but also introduce significant privacy concerns. The sensitive nature of gaze data can reveal personal information, such as emotional states, cognitive processes, and behavioral patterns [35, 4]. To address these challenges, scalable data management solutions and privacy-preserving techniques, such as secure real-time indexing and on-the-go encryption methods like homomorphic encryption [45], are essential for secure data analysis. Traditional approaches often struggle to meet the demands of scalability and real-time processing in XR usability studies, emphasizing the need for advanced, innovative solutions [36].

This paper addresses these challenges by leveraging the ELK stack [16], comprising Elasticsearch and Kibana, integrated within Unity, a leading game engine for XR development. Elasticsearch provides a scalable, distributed platform for managing large datasets, enabling near real-time indexing and query capabilities. Kibana offers powerful visualization tools for exploring trends and patterns, while Unity's scripting capabilities facilitate data collection, including gaze tracking and interaction metrics. Together, these tools support real-time analytics, scalability, and privacy, essential for usability studies in XR environments.

The integration of Elasticsearch with Unity offers several benefits:

- **Real-Time Analytics:** Dynamic monitoring of user interactions, enabling immediate experimental adjustments.
- **Scalability:** Efficient management of large datasets generated during prolonged or multi-user studies [13].
- **Comprehensive Insights:** Aggregation and analysis of complex datasets to uncover trends and behaviors.
- **Streamlined Studies:** Automated data collection and seamless processing for uninterrupted usability research.

This study investigates the integration of the Meta Quest Pro's gaze-tracking capabilities with the ELK stack in Unity. A preliminary study highlights both the opportunities and challenges of gaze-driven interactions in XR, underscoring the importance of balancing technological innovation with privacy considerations, such as serving as a potential testbed for encryption pipelines. These findings provide a foundation for developing secure, scalable, and immersive XR interfaces.

2 Related Work

Elasticsearch, the core component of the Elastic Stack (ELK), is a distributed search and analytics engine designed to handle large volumes of structured and unstructured data [10]. Elasticsearch offers unique advantages for XR usability studies, particularly its ability to handle high-volume, high-velocity data with real-time analytics. Alternatives like Apache Kafka [19], MongoDB [28], and PostgreSQL [20] each bring strengths but fall short in critical areas for this application. Kafka, while excellent for continuous event streaming, lacks built-in querying and analytics, requiring additional tools like Apache Druid, which increases complexity and latency. MongoDB handles unstructured data well but does not match Elasticsearch's advanced indexing and near-instantaneous query capabilities, particularly for real-time gaze metrics visualization. Similarly, PostgreSQL, though robust for structured data,

*e-mail: msunny@ualr.edu
†e-mail:jguthula@ualr.edu
‡e-mail:akharel@ualr.edu
§e-mail:mshraboni@ualr.edu
¶e-mail:harashid@ualr.edu
‖e-mail:pbhandari@ualr.edu
**email:jpspringer@ualr.edu
††e-mail:abasu@ualr.edu

struggles with the scalability and performance demands of high-frequency data ingestion, such as gaze tracking at 10ms intervals. In contrast, Elasticsearch combines scalable distributed indexing, rapid querying, and seamless visualization with Kibana, making it the more efficient and integrated choice for real-time XR applications. While Elasticsearch's resource requirements and learning curve pose challenges, its overall suitability for immersive environments justifies its use over these alternatives.

Persada et al. [33] demonstrated the capabilities of the ELK stack in analyzing sentiment data, while Ngo et al. [30] showcased Elasticsearch's potential for managing and visualizing complex datasets in real-time. These studies highlight the ELK stack's adaptability and scalability in addressing data management challenges, making it a suitable framework for usability studies in Unity. Data management challenges are not exclusive to VR environments. Similar challenges have been addressed in Operational Technology systems, where continuous data capture is critical due to the inability to take systems offline for forensics investigations. Akailvi et al. [2] presented an architecture leveraging existing technologies to enable continuous event capture in Operational Technology systems, highlighting its extensibility and application across diverse domains. Insights from these efforts underline the importance of scalable, real-time data management frameworks like Elasticsearch in solving complex analytical problems across fields. Since its release in 2010, Elasticsearch has evolved significantly, integrating features such as fault-tolerant shard architecture, near real-time indexing via inverted indexes, and support for both explicit and dynamic field mappings [31]. These advancements make Elasticsearch a scalable and flexible solution for data management, enabling its use in diverse domains, including real-time analytics in immersive environments.

In usability studies involving Unity-based Virtual Reality (VR) applications, data management presents significant challenges due to the high volume, variety, and velocity of interaction data generated. These datasets, which include motion tracking, eye tracking, and physiological metrics, require real-time integration, processing, and analysis [21]. Eye-tracking data, in particular, adds complexity, as it necessitates immediate processing to maintain immersion while addressing privacy concerns related to user focus, intent, and emotional states [31, 43]. Scalability is essential, as these studies often produce large datasets, especially in extended durations or multi-participant setups [42]. Tools like Elasticsearch provide a robust framework for managing and analyzing such data through efficient indexing, querying, and secure configurations, while Kibana enables actionable visualization of trends and insights [22, 7]. Addressing these challenges is critical for advancing usability studies in Unity-based VR environments.

The buzz-wire game serves as a focused testbed for evaluating gaze tracking, offering valuable insights into user attention, coordination, and decision-making processes. By analyzing how participants interact with this task, the study directly informs broader implications for XR usability, such as designing more intuitive interfaces and enhancing user engagement in immersive environments. This alignment ensures that the findings from the game translate meaningfully into actionable recommendations for real-world XR applications.

3 System Integration

We implemented the ELK Stack [16] to collect, analyze, and visualize real-time tracking data. The stack comprises Elasticsearch, Logstash, and Kibana, hosted in Linux containers for portability and efficient deployment. Using the Meta Quest Pro's integrated hand and eye-tracking sensors, gaze and hand movement data are captured and transmitted from Unity to Elasticsearch via its RESTful API. Transformation data of scene objects in Unity, recorded at a frequency of 10 ms, allows accurate replay of interactions. Additionally, system metrics such as frame rate (FPS), device name, operating system, and memory usage are logged to evaluate performance and behavior.

The ELK stack addresses these challenges by integrating three main components:

- **Elasticsearch:** A distributed search and analytics engine that stores and retrieves data at scale with near-instantaneous query speeds.
- **Logstash:** A data processing pipeline that ingests data from various sources, transforms it, and loads it into Elasticsearch.
- **Kibana:** A visualization tool that interfaces with Elasticsearch to create interactive dashboards and explore data trends.

In this study, we utilized Elasticsearch and Kibana as the core components of the ELK stack. Elasticsearch served as the primary engine for storing and querying the data, while Kibana provided intuitive dashboards for visualizing and analyzing the usability study data. These tools enabled seamless data management and insightful analysis, making them particularly effective for examining VR usability study data [25, 38].

The buzz-wire game serves as an ideal testbed for evaluating gaze tracking in XR environments due to its reliance on precision, spatial awareness, and real-time feedback. This task requires users to guide a virtual loop along a wire without making contact, demanding focused attention and deliberate hand-eye coordination. These characteristics directly align with gaze-specific metrics that are critical for understanding user behavior in immersive environments.

One key aspect of gaze tracking highlighted by this game is the analysis of fixation points. By examining where users fixate along the wire, researchers can identify areas that require higher cognitive effort or cause hesitation. Such insights are invaluable in understanding how users navigate challenging sections of the task and can inform design adjustments to enhance ease of use or engagement. The trajectory of gaze movements, or gaze pathways, further reveals the strategies users employ to complete the task. Some users may scan ahead along the wire to anticipate upcoming challenges, while others might fixate closely on the loop's immediate position for precise control. This data provides a window into individual differences in problem-solving and interaction styles within immersive environments. Sudden shifts in gaze direction, another critical metric, can indicate moments of uncertainty, distraction, or error. These moments provide clues about where users encounter difficulty or confusion, allowing developers to refine the interface or task design to better guide user behavior. Understanding attention shifts also helps to design interventions, such as visual cues, that can preemptively address user struggles.

By correlating these gaze metrics—fixation points, gaze pathways, and attention shifts—with task outcomes such as completion time and error frequency, the study demonstrates how gaze tracking can inform performance optimization in XR applications. The buzz-wire game's design inherently showcases the value of gaze tracking in identifying performance bottlenecks and enhancing usability. This connection underscores the broader applicability of gaze tracking in evaluating and improving interactive tasks within immersive environments.

3.1 Test Design

The VR application, inspired by the traditional buzz-wire game [12], involved guiding a virtual loop along a wire without making contact. The task commenced when any user initiated the session by pressing a button, which spawned a virtual loop in their hand.

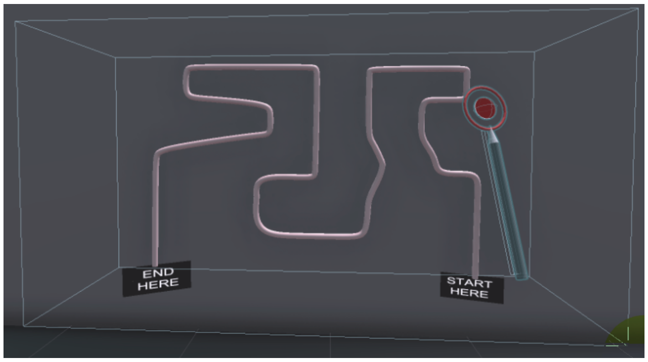


Figure 1: Virtual buzz-wire game setup. users guide a virtual loop along a wire using their chosen hand. The loop and its inner red circle turn black upon wire contact, offering real-time positioning feedback. The collider boundary (green lines) defines the designated *playing zone* in the VR task.

As users navigated the loop, visual feedback was provided for errors, with both the loop and an inner red circle turning black upon contact.

This VR implementation allowed for precise measurement of performance metrics, such as task completion time and contact frequency, surpassing the capabilities of the physical version. At the end of the task, users concluded the session by interacting with a reset button, which returned them to the entry scene. This setup supported real-time data collection and analysis while preserving the immersive experience (cf. Fig. 1).

3.2 Hardware Apparatus

Our study employed the Meta Quest Pro VR headset (Model: DK94EC) [9], featuring dual LCD displays with a resolution of 1832 x 1920 pixels per eye and a claimed field of view of 106° (horizontal) × 96° (vertical). The device was configured to operate at a refresh rate of 90 Hz to ensure smooth visual experiences. Equipped with the Qualcomm Snapdragon XR2+ processor, the Meta Quest Pro provides enhanced performance and supports advanced features such as built-in eye-tracking and precise motion tracking. The built-in eye-tracking feature was utilized to record gaze data, while the tracking capabilities enabled accurate assessment of user head and hand movements during the task. The setup was deployed in an approximately 9.29 m^2 area (10 ft. x 10 ft.) to facilitate natural interaction within the virtual environment.

3.3 Software

The VR application was developed using Unity 2022.3.54f1 with Oculus' **Movement SDK V71.0.1** [27], which integrates OpenXR APIs to enable Body Tracking (BT), Eye Tracking (ET), and Face Tracking (FT) features. This setup allowed the precise capture of gaze, hand, and body movements for real-time interaction and data logging.

3.4 Data Collection

Data collection was conducted using a Unity API, enabling the recording of spatial data points at intervals of 10 ms [8, 6]. The high-frequency data capture ensured precise tracking of various objects and movements within the virtual environment, including:

- Head movements, represented by the virtual camera.
- Eyegaze, represented by the eye tracker anchors present in OVR Camera Rig [27].
- Hand and loop movements, represented by the virtual loop.
- System metrics such as frame rate (FPS), device name, operating system, and memory usage.

In addition to spatial data, gaze tracking data (origin and direction) were recorded using the Meta Quest Pro's built-in sensors [5], while system performance metrics such as frame rate (FPS), device name, operating system, and memory usage were logged. These metrics provided a comprehensive dataset for both real-time and offline analysis.

Collision events were captured using strategically placed colliders [26] on key elements, such as the virtual wire, loop ring, and the designated *playing zone* and *wire zone* (cf. Fig. 1). Colliders were configured to align with the physical shapes of these objects, ensuring accurate detection of interactions. Binary flags tracked key states:

- The *in-wire zone* flag indicated whether the loop avoided wire contact.
- The *in-play zone* flag monitored whether users stayed within the designated area.

All captured data streams, including spatial data, gaze data, system metrics, and collision events, were logged locally and transmitted to an Elasticsearch instance in real-time via its RESTful API [15]. The data were indexed and stored for efficient querying and visualization in Kibana. This integration allowed for real-time analysis of user performance and interaction metrics, while the local storage provided a backup for offline evaluation. While the ELK stack facilitated seamless data ingestion and enabled the conversion of raw data into actionable insights, the focus of this implementation remains on system design and integration, specifically targeting usability metrics and interaction trends rather than in-depth behavioral analysis.

3.5 Elasticsearch on a HPC Cluster

The system architecture (cf. Fig. 2) integrates the ELK stack with a High-Performance Computing (HPC) environment to efficiently manage, analyze, and visualize high-volume interaction data generated during XR usability studies. An Elasticsearch cluster (version 8.17.0), consisting of three nodes deployed using lightweight Linux Containers (LXC) [18], ensures scalability, fault tolerance, and high availability. LXC provides resource isolation, efficiency, and security by sharing the host system's kernel, making it an optimal choice for resource-critical environments. A load balancer, configured with HAProxy in a round-robin setup [17], evenly distributes indexing and query requests across nodes, optimizing data ingestion and preventing bottlenecks. Additionally, the multi-node setup supports data replication, enhancing redundancy and ensuring that no data is lost even if a node fails.

Data collected via the `elk-logger` module, including eye-gaze tracking and motion metrics from devices like the Meta Quest Pro [9], is securely transmitted over a Wide Area Network (WAN) to the Elasticsearch cluster for real-time storage, indexing, and querying [1]. To safeguard data privacy, all communications between the Elasticsearch nodes, Kibana, `elk-logger`, and `elk-replay` modules are encrypted using SSL/TLS protocols. The `elk-replay` module retrieves the indexed data to replay sessions within a Unity-based XR environment, enabling researchers to assess user behavior and interaction history [13].

Visualizations and analytics are facilitated through Kibana, which provides intuitive dashboards for real-time user metrics, including gaze focus, task accuracy, and system performance [21, 29]. While this study does not implement machine learning jobs within Elasticsearch, the architecture depicted in Fig. 2 demonstrates the potential placement of machine learning components, highlighting future possibilities for advanced analytics (cf. Sec. 6.1).

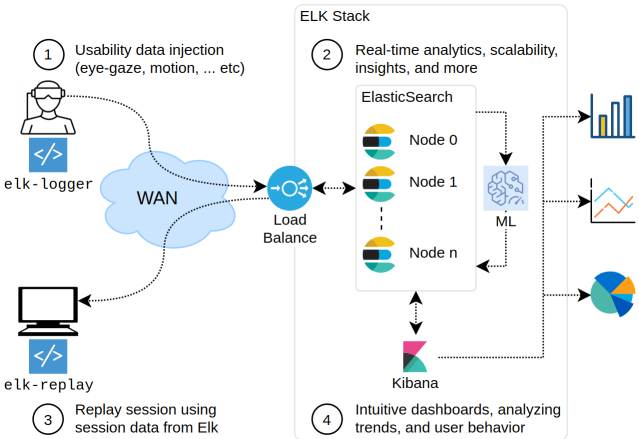
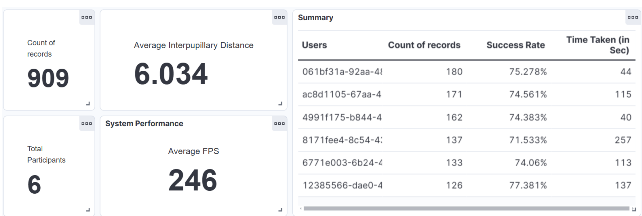


Figure 2: System architecture for integrating the ELK stack into XR usability studies. This setup supports real-time data ingestion, storage, analytics, and visualization.



Users	Count of records	Success Rate	Time Taken (in Sec)
061bf31a-92aa-4	180	75.278%	44
ac8d1105-67aa-4	171	74.561%	115
4991f175-b844-4	162	74.383%	40
8171fee4-8c54-4	137	71.533%	257
6771e003-6b24-4	133	74.06%	113
12385566-dae0-4	126	77.381%	137

Figure 3: Sample Kibana Dashboard Displaying Real-Time Usability and System Performance Metrics”

4 Design Evaluation

The experiment’s data visualization makes use of ELK Stack and Unity to offer a variety of techniques for analyzing the stored data. Kibana, a component of the ELK Stack, is used as a non-immersive tool for aggregating and showing key metrics. In contrast, Unity is used as an immersive platform to replay user sessions and interactively analyze gaze data.

4.1 Kibana Analytics

Kibana offers interactive dashboards (cf. Fig. 3) that consolidate real-time metrics essential for evaluating user performance and system behavior during usability studies. These dashboards include parameters such as user count, total data records, eye-gaze metrics, task accuracy, and frames per second (FPS), providing a comprehensive overview of the study’s status. User count reflects active user engagement, while total data records monitor the volume of collected interaction data, ensuring the data capture process is functioning optimally.

$$A = \frac{\sum_{i=1}^{N} \text{flag}_i}{N_{\text{wire zone}}} \quad (1)$$

Task accuracy, defined as the success rate during the task, is computed as the ratio of instances where the ring did not make contact with the wire (*in-wire zone flag = true*) to the total number of data points recorded while the ring was within the wire zone, as expressed in Eq. (1). This metric evaluates users’ precision and consistency in task performance [16, 13]. Eye-gaze metrics provide insights into user focus, revealing interaction patterns and behavior, while FPS logs ensure system performance remains optimal, delivering a smooth and immersive experience.

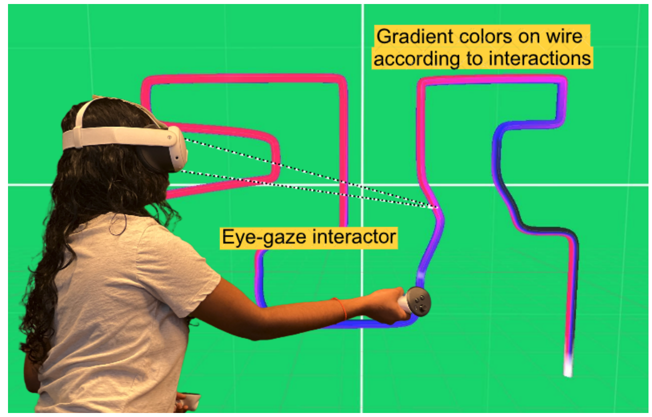


Figure 4: Heatmap Visualization in Unity Highlighting User Focus Based on Gaze-Tracking Data from Elasticsearch.

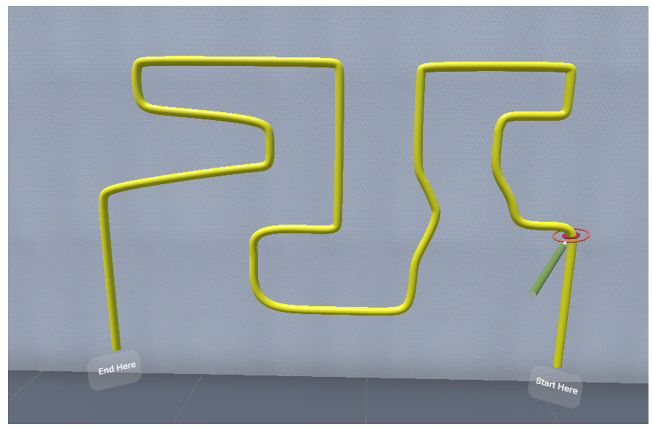


Figure 5: Reconstructed a replay scene in Unity utilizing data retrieved from Elasticsearch.

This visualization framework integrates key usability metrics and system performance indicators, facilitating dynamic monitoring, real-time adjustments, and a detailed analysis of study outcomes.

4.2 Replay Sessions

Unity enables an advanced replay system by leveraging transformation data (position, rotation, and scale) stored in Elasticsearch to recreate user interactions within the virtual environment (cf. Fig. 5). The replay algorithm queries Elasticsearch for session data using a RESTful API, parses the response, and organizes the data into chronological frames for playback. Each frame captures the state of game objects, and the replay iterates through frames using interpolation to ensure smooth transitions. Missing objects are dynamically instantiated to maintain scene consistency, with options for visual enhancements like heatmap overlays to analyze user focus areas. This approach provides a robust, scalable solution for reviewing and analyzing interaction data, supporting iterative improvements in XR usability studies.

4.3 Heatmap Analytics

We implemented a heatmap visualization system in Unity (c.f Fig. 4) to intuitively represent user focus points during XR tasks. This system utilizes raycasting originating from the user’s gaze direction, retrieved from Elasticsearch, to map focus areas dynamically within the virtual environment. Scene objects are dynamically configured to support heatmap generation by ensuring appropriate colliders and materials are applied, while non-Mesh colliders and other irrelevant components are removed to optimize performance [34]. A specialized heatmap shader visualizes the intensity of user attention by highlighting areas where gaze rays intersect with objects, offering a clear and intuitive representation of user focus throughout the task. This approach provides valuable insights into interaction patterns, enabling iterative improvements to XR applications.

5 Discussion

Our integration of Elasticsearch with Unity provides significant advantages for usability studies, enhancing efficiency in data collection and analysis. Incorporating a multi-node cluster in the ELK setup improves fault tolerance, load balancing, and scalability, ensuring consistent performance under high data loads and enabling seamless ingestion and querying of large XR datasets [16, 13].

This implementation focuses on system design and integration, targeting usability metrics and interaction trends rather than in-depth behavioral analysis, which remains beyond the scope of this work. By enabling real-time analytics, the system dynamically monitors user interactions, supports immediate experimental adjustments, and aggregates complex datasets to uncover trends and patterns. Future iterations could leverage Elasticsearch's built-in integrations and advanced privacy-preserving techniques, such as homomorphic encryption, to further enhance secure and scalable XR usability research.

The integration of Elasticsearch with Unity provides significant advantages for XR usability studies, enhancing real-time data collection, indexing, and visualization. However, the system is not without its limitations. One key challenge lies in the scalability of the framework when applied to larger-scale or prolonged studies. While Elasticsearch is optimized for high-velocity data, the computational resources required for real-time indexing and querying, particularly in multi-node setups, may become prohibitive in resource-constrained environments. Additionally, the framework's reliance on the Meta Quest Pro headset introduces hardware dependency, which may limit its generalizability across XR platforms with differing capabilities.

Another limitation is the potential for biases in data collection. Factors such as participant variability, environmental conditions, or inaccuracies in gaze-tracking sensors could skew the results, affecting the reliability of insights. Future iterations of the system should incorporate calibration mechanisms and advanced error-correction algorithms to minimize these biases.

Despite these limitations, the framework offers exciting opportunities for extension. One potential direction is adapting the system for multi-user environments, enabling simultaneous data collection and analysis across multiple participants in collaborative XR tasks. This extension could provide deeper insights into group dynamics and shared decision-making in immersive environments. Another avenue is the integration of additional biometric sensors, such as EEG or heart rate monitors, to correlate gaze data with physiological states. This combination would offer a more holistic understanding of user experiences, particularly in applications requiring cognitive or emotional assessment.

Ethical considerations remain paramount in the deployment of gaze-tracking technologies. In commercial settings, the use of gaze data for targeted advertising or behavioral analysis raises concerns about user consent and data exploitation. Similarly, in healthcare applications, where gaze data could reveal cognitive or emotional conditions, safeguarding sensitive information is critical to maintaining patient trust. The system's use of privacy-preserving mechanisms, such as homomorphic encryption, represents a significant step toward addressing these concerns, but ongoing efforts are needed to align with evolving regulations and ethical standards.

6 Future Implementation

6.1 Machine Learning Integration with Elasticsearch

Integrating machine learning (ML) with the Elastic Stack in XR usability studies offers advanced analytics capabilities, such as anomaly detection and real-time user adaptation, but presents several challenges. Data quality and preparation are significant hurdles, as raw XR data often contain noise, inconsistencies, or missing values. Developing a robust preprocessing pipeline for real-time requirements is essential but complex.

Real-time ML tasks like gaze pattern clustering and behavior prediction require substantial computational resources. While Elasticsearch supports feature extraction and basic analytics, integrating more advanced models from frameworks like TensorFlow or PyTorch can increase latency and strain system performance. Additionally, training and deploying these models for diverse user populations demand extensive datasets and computationally intensive workflows, often requiring HPC or edge infrastructure for efficiency.

Privacy concerns further complicate ML integration, as gaze data can reveal sensitive information about users' emotional states or cognitive processes. Compliance with privacy regulations such as General Data Protection Regulation (GDPR) [32] or California Consumer Privacy Act (CCPA) [24] necessitates techniques like federated learning or differential privacy, which introduce additional complexity. Scalability also poses challenges, especially in multi-user environments where the volume and complexity of data can overwhelm the system.

To address these challenges, future implementations should focus on lightweight ML models for real-time analytics, with advanced processing reserved for offline tasks. Hybrid approaches combining Elasticsearch's native ML tools with external frameworks can balance simplicity and functionality. Privacy-preserving methods and scalable architectures, such as containerized systems, will be critical for ensuring ethical and efficient deployments.

By tackling these challenges, ML-integrated systems can unlock deeper, more actionable insights from gaze-driven data, significantly enhancing XR usability studies and applications.

6.1.1 Synthetic Data Generation

Elasticsearch's ML capabilities can generate synthetic datasets for XR usability studies by querying indexed gaze-tracking and interaction data, replacing sensitive PII with anonymized or artificial data to ensure privacy while maintaining usability. ML APIs such as `_ml/anomaly_detectors` and `_ml/data_frame_analytics` streamline integration into XR workflows, enabling anomaly detection and predictive analytics on user behavior [14]. Challenges like model biases and non-representative patterns can be mitigated through rigorous validation and feedback loops [21, 29].

6.1.2 Decentralized Learning and HPC Integration

In the context of XR usability studies, Elasticsearch's ML capabilities enable synthetic data generation by querying and anonymizing sensitive gaze-tracking and interaction data. Using the `POST _ml/data_frame/analytics/` endpoint, feature extraction, anomaly detection, and clustering can be performed to gain advanced insights while preserving privacy [14]. This approach ensures efficient management of large datasets through Elasticsearch's scalability and facilitates seamless integration with high-performance computing (HPC) clusters for distributed data processing. Challenges like model biases and non-representative patterns can be mitigated through iterative validation, enhancing both the privacy and usability of generated data [21, 29].

6.1.3 Machine Learning Pipelines with the ELK Stack

The ELK stack offers an efficient framework for building machine learning (ML) pipelines in XR usability studies by analyzing user behavior, predicting outcomes, and refining immersive environments. Elasticsearch's aggregation tools extract meaningful features like task duration, interaction frequency, and gaze focus, aiding in understanding user behavior [1, 13, 44, 4, 11]. Built-in capabilities support anomaly detection, regression, and classification, enabling the identification of navigation challenges [16, 21, 29]. Kibana visualizes ML outputs, providing real-time feedback for iterative model refinement [42]. Although High-Performance Computing (HPC) clusters can enhance scalability, server-based uploads ensure efficient, real-time data handling for practical deployments [31, 30].

6.2 Homomorphic Encryption for XR Data Security

Usability studies in immersive environments generate vast amounts of sensitive data, such as gaze patterns and interaction logs, which pose significant privacy risks. Traditional encryption methods often fall short in XR applications where real-time processing is essential. Homomorphic Encryption (HE) addresses these challenges by enabling computations on encrypted data without the need for decryption. This ensures that data remains secure throughout the processing workflow, significantly reducing exposure risks [40, 23].

In our future implementation, we aim to integrate HE into the ELK stack to securely analyze sensitive fields such as gaze origin, gaze direction, and interaction zones. By encrypting these fields at the point of collection, we ensure they remain protected during aggregation and querying processes in Elasticsearch. This allows metrics such as gaze density, fixation duration, and interaction heatmaps to be computed without exposing raw data at any stage. Tools like Microsoft SEAL [41] provide the necessary framework for implementing efficient encryption and querying while maintaining the real-time performance critical to XR applications.

HE offers several advantages over conventional encryption methods. By eliminating the need for plaintext data during analysis, HE contributes to meeting the data security requirements outlined in privacy regulations such as the GDPR [32] and the CCPA [24]. Furthermore, HE supports scalable and secure data handling for XR environments, making it ideal for real-time operations where performance is critical. Theoretical models presented in [3] further validate the feasibility of integrating HE within the ELK stack, enabling privacy-preserving visualizations of encrypted data.

While we are in the process of exploring the use of HE for analyzing encrypted data in similar data-intensive contexts, this work remains a work-in-progress and is currently unpublished. Our aim is to extend this approach to sensitive XR data within the ELK stack. By encrypting fields such as gaze direction and interaction zones at the point of collection, we plan to secure sensitive information throughout the analysis pipeline. This envisioned system would compute metrics such as gaze density and interaction heatmaps without requiring decryption, preserving privacy while providing actionable insights. Kibana dashboards would then visualize these encrypted analytics, ensuring user data confidentiality at every stage of processing.

7 Conclusions

This study highlights the integration of Elasticsearch and Kibana within Unity-based XR environments, presenting a comprehensive framework for the management, analysis, and visualization of gaze-tracking and interaction data. By addressing critical challenges in scalability, real-time analytics, and privacy preservation, the system significantly enhances the efficiency and robustness of usability studies while ensuring the secure handling of sensitive data. The integration of machine learning pipelines offers potential for sophisticated anomaly detection, predictive modeling, and real-time system adaptability, further augmenting the analytical capabilities available to researchers. Additionally, privacy-preserving methodologies, such as homomorphic encryption, demonstrate a forward-thinking approach to safeguarding sensitive gaze-tracking data.

The practical implications of this study extend to various industries. In gaming, the system can be used to design more engaging and adaptive player experiences by analyzing gaze patterns and interaction behaviors. In healthcare, it holds potential for patient monitoring and cognitive assessments, offering new avenues for diagnosis and treatment planning. In education, gaze-based analytics could be employed to evaluate student engagement and optimize learning materials in immersive training environments.

The system also demonstrates scalability and adaptability for diverse XR applications. Its modular design allows for seamless integration with multi-user environments, collaborative XR tasks, and additional biometric sensors, making it suitable for a wide range of research and commercial scenarios. By leveraging the Elastic Stack's inherent flexibility, the framework can be tailored to meet the specific requirements of different domains.

Finally, this study underscores the broader impact of advancing ethical data usage in immersive technologies. As gaze tracking becomes increasingly prevalent, ensuring the privacy and security of user data is critical to maintaining trust and promoting equitable adoption of these technologies. This work lays the groundwork for future innovations that prioritize both functionality and ethics, positioning gaze-driven analytics as a cornerstone of next-generation XR applications.

References

[1] D. D. Ahir and N. F. Shaikh. Evaluation of elasticsearch ecosystem including machine learning capabilities. *International Journal of Safety & Security Engineering*, 14(4), 2024. 3, 6

[2] S. Akailvi, U. Gautam, P. Bhandari, H. Rashid, P. D. Huff, and J. P. Springer. Helot–hunting evil life in operational technology. *IEEE Transactions on Smart Grid*, 14(4):3058–3071, 2022. 2

[3] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan. Homomorphic encryption standard. Cryptology ePrint Archive, Paper 2019/939, 2019. 6

[4] Y. Banchi, Y. Ohira, M. Ito, and T. Kawai. Comparative analysis of user experience in virtual reality (vr) and mixed reality (mr) systems using eye-tracking measurements. *Electronic Imaging*, 36:1–4, 2024. 1, 6

[5] W. Barbel. Neurogaze in virtual reality: Assessing an eeg and eye tracking interface against traditional virtual reality input devices. 2024. 3

[6] A. Basu. Stag: A tool for realtime replay and analysis of spatial trajectory and gaze information captured in immersive environments. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 43–45. IEEE, 2022. 3

[7] D. Bordoloi and S. Shukla. An elk stack method with machine learning algorithm for alerting traffic anomaly. *Webology*, 18(5):3228–3234, 2021. 2

[8] J. Bueno. Development of unity 3d module for rest api integration: Unity 3d and rest api technology. 2017. 3

[9] K. Chen, T. Wan, N. Matsuda, A. Ninan, A. Chapiro, and Q. Sun. Pea-pods: Perceptual evaluation of algorithms for power optimization in xr displays. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024. 1, 3

[10] K. Y. Cheng, S. Pazmino, H. Qian, B. Bergh, and B. Schreiweis. Structured data retrieval and analysis of hl7 v2 messages with elasticsearch. In *MEDINFO 2023—The Future Is Accessible*, pp. 1464–1465. IOS Press, 2024. 1

[11] G. R. Chhimpa, A. Kumar, S. Garhwal, and Dhiraj. Empowering individuals with disabilities: a real-time, cost-effective, calibration-free assistive system utilizing eye tracking. *Journal of Real-Time Image Processing*, 21(3):97, 2024. 6

[12] C. G. Christou, D. Michael-Grigoriou, and D. Sokratous. Virtual buzzwire: Assessment of a prototype vr game for stroke rehabilitation.

In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 531–532. IEEE, 2018. 2

[13] K. Elansari, A. Idrissi, and K. Moutaouakil. Elk stack approach with artificial intelligence for logs collection and resource usage monitoring and forecasting. In *Modern Artificial Intelligence and Data Science 2024: Tools, Techniques and Systems*, pp. 515–527. Springer, 2024. 1, 3, 4, 5, 6

[14] Elastic. Machine learning api quick reference, 2023. Accessed: 2024-12-21. 5

[15] Elastic. Rest apis — elasticsearch guide [current], 2024. Accessed: 2024-12-21. 3

[16] Elastic Community. Elasticsearch Guide. Retrieved from `https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html`, 2024. 1, 2, 4, 5, 6

[17] H. Z. Fahmi, I. Buditjahjanto, A. Hafidz, D. A. Dermawan, I. Sidhimantra, and R. N. Saputra. Optimizing resource availability on high scalability with container technology through a load balancing approach. In *AIP Conference Proceedings*, vol. 3116. AIP Publishing, 2024. 3

[18] M. Farahmandian, M. F. Foumani, and P. Bayat. Improving fault tolerance in linux container-based distributed systems using blockchain. *Cluster Computing*, pp. 1–10, 2024. 3

[19] A. S. Foundation. Apache kafka: A distributed streaming platform, 2025. `https://kafka.apache.org/`. 1

[20] P. G. D. Group. Postgresql: The world's most advanced open source relational database, 2025. `https://www.postgresql.org/`. 1

[21] M. T. Islam and B. U. Khan. Big data and analytics: Prospects, challenges, and the way forward. In *Encyclopedia of Information Science and Technology, Sixth Edition*, pp. 1–30. IGI Global, 2025. 2, 3, 5, 6

[22] J. S. Lau, Y. M. Tang, G. Gao, K. N. Fong, and B. C. So. Development and usability testing of virtual reality (vr)-based reminiscence therapy for people with dementia. *Information Systems Frontiers*, pp. 1–16, 2024. 2

[23] K. Lauter, W. Dai, and K. Laine. Protecting privacy through homomorphic encryption. 2021. 6

[24] C. Legislature. California consumer privacy act (ccpa), 2018. Accessed: 2025-01-06. 5, 6

[25] Y.-E. Lu, S.-C. Yeh, E. H.-K. Wu, L. Liu, and S. K. Lin. virtual reality based neural sensors with respirator for cardiorespiratory synchrony training system. In *2024 9th IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 86–91. IEEE, 2024. 2

[26] E. Mendeleev, D. Burtseva, Y. Kiliba, R. Petrov, S. Bozhkov, I. Milenov, and P. Bozhkov. Virtual design of physical processes. In *2022 22nd International Symposium on Electrical Apparatus and Technologies (SIELA)*, pp. 1–4. IEEE, 2022. 3

[27] I. Meta Platforms. Meta horizon move overview for unity. `https://developers.meta.com/horizon/documentation/unity/move-overview/`, 2024. Accessed: 2024-12-27. 3

[28] I. MongoDB. Mongodb: The developer data platform, 2025. `https://www.mongodb.com/`. 1

[29] J. Moon, M. Jeong, S. Oh, T. H. Laine, and J. Seo. Data collection framework for context-aware virtual reality application development in unity: case of avatar embodiment. *Sensors*, 22(12):4623, 2022. 3, 5, 6

[30] T. T. T. Ngo, D. Sarramia, M.-A. Kang, and F. Pinet. A new approach based on elk stack for the analysis and visualisation of geo-referenced sensor data. *SN Computer Science*, 4(3):241, 2023. 2, 6

[31] H. Nolte, L. Quentin, and J. Kunkel. Secure elasticsearch clusters on hpc systems for sensitive data. In *International Conference on High Performance Computing*, pp. 199–212. Springer, 2025. 2, 6

[32] E. Parliament. Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, 59:1–88, 2016. 5, 6

[33] S. Persada, A. Oktavianto, B. Miraja, R. Nadlifatin, P. Belgiawan, and A. P. Redi. Public perceptions of online learning in developing countries: A study using the elk stack for sentiment analysis on twitter. *International Journal of Emerging Technologies in Learning (iJET)*, 15(9):94–109, 2020. 2

[34] P. Pieta, H. Jegierski, P. Babiuch, M. Jegierski, M. Płaza, G. Łukawski, S. Deniziak, A. Jasiński, J. Opałka, P. Wegrzyn, et al. Automated classification of virtual reality user motions using a motion atlas and machine learning approach. *IEEE Access*, 2024. 5

[35] T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE symposium on 3D user interfaces (3DUI)*, pp. 36–39. IEEE, 2017. 1

[36] T. P. Raptis, A. Passarella, and M. Conti. Data management in industry 4.0: State of the art and open challenges. *IEEE Access*, 7:97052–97093, 2019. 1

[37] M. Sanaei, S. B. Gilbert, N. Javadpour, H. Sabouni, M. C. Dorneich, and J. W. Kelly. The correlations of scene complexity, workload, presence, and cybersickness in a task-based vr game. In *International Conference on Human-Computer Interaction*, pp. 277–289. Springer, 2024. 1

[38] A. S. Shaji and M. M. George. Elastic stack: A comprehensive overview. *2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 1–5, 2024. 2

[39] M. J. M. Sunny and A. Basu. Non-linear parameterization of spatial decision making in immersive virtual environment. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 389–395, 2024. doi: 10.1109/VRW62533.2024.00076 1

[40] A. M. L. R. Team. Combining machine learning and homomorphic encryption in the apple ecosystem, 2023. 6

[41] M. R. Team. Microsoft seal: Fast and easy-to-use homomorphic encryption library. 6

[42] L. Theodorakopoulos, A. Theodoropoulou, and Y. Stamatiou. A state-of-the-art review in big data management engineering: Real-life case studies, challenges, and future research directions. *Eng*, 5(3):1266–1297, 2024. 2, 6

[43] S. Van Damme, J. Sameri, S. Schwarzmann, Q. Wei, R. Trivisonno, F. De Turck, and M. Torres Vega. Impact of latency on qoe, performance, and collaboration in interactive multi-user virtual reality. *Applied Sciences*, 14(6):2290, 2024. 2

[44] C.-T. Yang, Y.-W. Chan, J.-C. Liu, E. Kristiani, and C.-H. Lai. Cyberattacks detection and analysis in a network log system using xgboost with elk stack. *Soft Computing*, 26(11):5143–5157, 2022. 6

[45] X. Yi, R. Paulet, E. Bertino, X. Yi, R. Paulet, and E. Bertino. *Homomorphic encryption*. Springer, 2014. 1