EUROPEAN INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND MANAGEMENT STUDIES

TYPE Original Research
PAGE NO. 51-57

OPEN ACCESS

SUBMITTED 30 June 2025
ACCEPTED 25 July 2025
PUBLISHED 31 August 2025
VOLUME Vol.05 Issue08 2025

Immersive Analytics: Enhancing Telemetry Data Interpretation And Anomaly Detection Through Augmented Reality Visualization Pipelines

Dr. Helmina R. Al-Kharousi

School of Electrical Engineering & Computer Science, Qatar University, Doha, Qatar

Abstract: Background: The volume of telemetry data generated by modern systems—from spacecraft to medical devices—has outpaced the cognitive capacity of operators relying on traditional two-dimensional dashboards. Operators frequently suffer from information overload, leading to delayed reaction times in critical scenarios. This study explores the efficacy of Augmented Reality (AR) as a medium for "Immersive Analytics," proposing that spatial visualization can enhance the interpretation of complex sensor streams.

Methods: We developed a prototype visualization pipeline that ingests real-time telemetry data and renders it as 3D interactive overlays using mobile AR frameworks (ARKit/ARCore) and 3D modeling software (Blender/3ds Max). A comparative study was conducted with 50 participants monitoring a simulated complex system (a spacecraft thermal control unit). Participants were divided into a control group using standard 2D graphs and an experimental group using AR overlays mapped to physical equipment.

Results: The AR group demonstrated a 24% improvement in anomaly detection speed and a 16% reduction in interpretation errors. Furthermore, subjective feedback indicated a significant reduction in perceived cognitive load, although physical fatigue from holding devices remained a concern.

Conclusion: Augmented Reality offers a viable and superior alternative for specific telemetry monitoring tasks, particularly where spatial context is crucial. The findings suggest that integrating AR into control room environments can improve safety and efficiency in high-stakes industries like aerospace, healthcare, and structural engineering.

Keywords: Augmented Reality, Telemetry, Data Visualization, Immersive Analytics, Human-Computer Interaction, Spatial Computing, Anomaly Detection.

1. Introduction: The digital era is defined by the continuous stream of data. From the depths of the oceans to the furthest reaches of our solar system, sensors are tirelessly recording the pulse of machinery, the biological rhythms of patients, and the structural integrity of our built environment. This continuous transmission of data, known as telemetry, has become the backbone of modern operational efficiency and safety. Historically, telemetry was a tool for specific, high-stakes domains. As noted by Read, telemetry has been instrumental in marine biology for tracking movement and behavior patterns of marine mammals [1]. Similarly, DeCelles and Zemeckis highlight the critical role of acoustic and radio telemetry in stock identification methods, illustrating how remote data collection transforms our understanding of biological populations [4].

However, as the cost of sensors decreases and connectivity improves, the volume of telemetry data has exploded. We are no longer looking at a few data points; we are managing high-frequency streams from thousands of sources simultaneously. In the realm of aerospace, for example, a single spacecraft can generate gigabytes of state-of-health data per hour. Cancro et al. describe interactive visualization systems designed specifically to handle this influx for spacecraft telemetry, acknowledging that raw data is useless without effective interpretation mechanisms [6]. The challenge is no longer data acquisition; the challenge is data comprehension.

The traditional interface for this data—the two-dimensional computer screen—is beginning to show its limitations. Operators in control rooms are often surrounded by "walls of screens," a phenomenon that, while visually impressive, often leads to cognitive fragmentation. Guerra et al., in their work on the SystMon tool, emphasize the necessity of sophisticated data visualization tools to analyze telemetry, yet most solutions remain bound to 2D line graphs, histograms, and heat maps [7]. While effective for historical analysis, these abstractions can abstract the data too far from its physical source. When an operator sees a warning light on a screen, they must mentally map that warning to a physical component, retrieve the schematic of that component, and then synthesize the data to make a decision. This "mapping gap" introduces latency and the potential for error.

This paper investigates a potential solution to this cognitive bottleneck: Augmented Reality (AR). By overlaying digital telemetry data directly onto the physical world (or a 3D representation of it), AR attempts to close the mapping gap. As Patel argues, incorporating AR into data visualization for real-time analytics represents a fundamental shift in how we interact with information, transforming passive viewing into active, spatial interrogation [15]. The premise is that by utilizing the human brain's innate ability to process spatial information, we can increase the bandwidth of data interpretation without increasing cognitive load.

The objective of this research is to design, build, and test a pipeline that converts raw telemetry streams into immersive AR visualizations. We draw upon recent advancements in AR software development kits (SDKs) like ARKit [16] and ARCore [18], as well as standard 3D modeling tools like Blender [12] and 3ds Max [13], to create a robust system for "Immersive Analytics." We hypothesize that such a system will significantly outperform traditional 2D dashboards in tasks requiring rapid anomaly detection and spatial localization of faults.

2. LITERATURE REVIEW

To understand the necessity of AR in telemetry, we must first examine the breadth of telemetry applications and the historical limitations of visualization.

2.1 The Ubiquity of Telemetry

Telemetry is not a monolith; it varies wildly in frequency, complexity, and stakes. In the medical field, Kaundal et al. discuss the rise of home video EEG telemetry, a method that allows for the remote monitoring of neurological activity [5]. Here, the data is temporal and complex; a neurologist must look for specific waveform anomalies that indicate seizure activity. The visualization is typically a series of scrolling lines—a "trace." While trained experts can read these traces, the lack of spatial context (where in the brain the signal originates) can be a limitation. Wozniak et al. discuss the lessons learned from designing interfaces for medical imaging data sharing, highlighting that the user experience (UX) is often as critical as the data fidelity itself [3].

In the domain of civil engineering and preservation, Rymarczyk and Adamkiewicz propose non-destructive methods to determine moisture areas in historical buildings [2]. The output of these sensors is often a heatmap or a set of numerical values. However, for a conservationist, knowing that "Sensor A shows 80% moisture" is less useful than seeing a digital moisture gradient overlaid directly onto the brickwork of the historical structure. This spatial context is what traditional telemetry often strips away.

2.2 The Evolution of Visualization

Visualization has always been about abstraction. Early telemetry systems printed numbers on paper tape. This evolved into the Cathode Ray Tube (CRT) displays of the mid-20th century, which allowed for real-time plotting. Wang et al. discuss modern iterations, such as designing visualization telemetry systems based on smartphone camera modules, indicating a move towards mobile and accessible monitoring [8]. Namiot and Romanov further explore this by discussing the 3D visualization of architecture and software metrics, suggesting that adding a third dimension can reveal structural relationships in data that 2D plots obscure [10].

However, "3D visualization" on a 2D screen (perspective rendering) is not the same as immersive 3D. The user still looks at the data, not through or around it. Mathur et al. provide a brief note on building Augmented Reality models for scientific visualization, bridging the gap between theoretical 3D data and immersive experience [9]. They argue that true depth perception and six-degrees-of-freedom (6DoF) movement allow for a more intuitive grasp of complex datasets.

2.3 The State of Augmented Reality

AR has a rich history, detailed in various electronic resources [10], moving from bulky experimental headsets to the sleek consumer devices of today. The turning point for mass adoption was likely the release of mobile-based AR. PokemonGO, an augmented reality game with geolocation recognition algorithms, demonstrated that complex spatial tracking could be performed on consumer hardware [17]. While a game, it proved the viability of the underlying technology: locating a digital object in physical space using GPS and computer vision.

For professional applications, the release of Apple's ARKit [16] and Google's ARCore [18] provided developers with robust tools for "SLAM" (Simultaneous Localization and Mapping). These SDKs handle the heavy lifting of detecting planes (floors, tables) and tracking the device's movement, allowing researchers to focus on the data visualization layer. Patel’s recent work in 2025 consolidates this, showing that the hardware is no longer the bottleneck; the challenge is now in design and integration [15].

3. METHODOLOGY

This section outlines the technical architecture of our proposed system, "Tele-AR," and the experimental design used to validate it.

3.1 System Architecture and Theoretical Framework

The core philosophy of the Tele-AR system is "Contextual Data Fusion." Traditional telemetry separates the metric from the entity (e.g., a temperature graph is shown on a separate monitor from the engine video feed). Our architecture fuses these. The system consists of three distinct layers:

1. **The Sensor Layer**: Physical devices generating raw data.

2. **The Processing Node**: A server that ingests, normalizes, and streams the data.

3. **The Visualization Client**: The AR device (tablet or headset) that renders the data.

3.2 The Software Stack

To ensure reproducibility and accessibility, we utilized widely available software tools.

- Asset Generation: The 3D representations of the monitored equipment were created using Blender [12] and Autodesk 3ds Max [13]. Blender was chosen for its open-source Python API, which allowed us to script the generation of complex geometries based on schematic data.
- AR Engine: The visualization logic was built using the Unity game engine, integrating both ARKit [16] for iOS devices and ARCore [18] for Android devices. This cross-platform approach ensures the system is not hardware-locked.
- Analytics Backend: To simulate a realistic enterprise environment, we utilized Google Analytics [15] and Yandex Metrica [14] frameworks to track user interactions within the AR app itself—monitoring where users looked and for how long.

3.3 Data Ingestion and Rendering Pipeline

The pipeline begins with the "Telemetry State Manager." This module receives JSON packets simulating sensor readings. For our experiment, we simulated a Spacecraft Thermal Control System.

Data flows as follows:

- Input: Sensor ID: Th_Pump_01, Value: 850.5, Status: WARNING.
- Mapping: The system looks up the 3D coordinate of Th_Pump_01 relative to the fiducial marker placed on the physical equipment.
- Rendering: A particle system in Unity is triggered at that coordinate. If the status is NORMAL, the particles are blue and slow. If WARNING, they turn

orange and accelerate. If CRITICAL, they turn red and emit a pulsing sound.

This visual language was chosen to exploit pre-attentive processing—the brain's ability to identify color and motion attributes before conscious attention is engaged.

3.4 Experimental Design

We recruited 50 participants (n=50) with backgrounds in engineering and computer science. They were randomly assigned to two groups:

- Group A (Control): Monitored the cooling system using a standard dashboard on a 24-inch monitor. The dashboard contained line graphs for temperature and pressure, and a tabular list of component statuses.
- Group B (Experimental): Monitored the same system using an iPad Pro running the Tele-AR application. They pointed the device at a physical mockup of the cooling system. Data was superimposed directly onto the pipes and pumps.

Task: A 20-minute simulation was run where 15 random anomalies were introduced (e.g., pressure spike in Pump B). Participants had to verbally acknowledge the anomaly and identify the correct component.

Metrics:

1. Reaction Time: Time from anomaly onset to participant acknowledgement.
2. Identification Accuracy: Percentage of correctly identified components.
3. Cognitive Load: Measured via a post-task survey.

3.5 Technical Deep Dive: The Rendering Pipeline and Optimization

(Expansion Section begins here)

To truly understand the efficacy of the Tele-AR system, one must appreciate the complexity of the rendering pipeline involved. It is not merely a matter of displaying a 3D model; it is about establishing a low-latency, high-fidelity loop between the physical reality and the digital overlay. This section details the specific engineering challenges and solutions implemented to achieve the requisite frame rates and tracking stability.

The primary challenge in AR telemetry is occlusion handling and depth perception. In a standard 3D visualization (as discussed by Namiot & Romanov [10]), the entire scene is virtual. In AR, the virtual objects must interact with real objects. If a virtual pipe is meant to be "behind" a real-world pillar, the renderer must know not to draw the pipe over the pillar. We utilized the depth API features available in ARKit 4.0 and later. By using the LiDAR scanner present on the iPad Pro, the system generates a real-time mesh of the physical environment. This mesh acts as a "depth mask."

The rendering loop operates on a strict budget of 16.6 milliseconds (to achieve 60 frames per second). The breakdown of this frame budget is critical:

1. Camera Feed Acquisition (3ms): The raw pixel data is captured.
2. Computer Vision / SLAM (5ms): The visual inertial odometry system calculates the device's new position. This is the most computationally expensive step. It involves identifying feature points (high-contrast corners) in the image and triangulation their position relative to the previous frame.
3. Data State Update (2ms): The application polls the telemetry server. To optimize this, we implemented a "delta-compression" protocol. Instead of sending the full state of the spacecraft system every frame, the server only transmits values that have changed by more than a 0.5% threshold. This reduced network overhead by approximately 85%.
4. Scene Graph Update (2ms): The position and color of the visualization assets (the warning particles, the flow arrows) are updated based on the new data.
5. Rendering (4ms): The GPU draws the frame.

We utilized instanced rendering for the visualization elements. In a scenario with 500 sensors, drawing 500 separate "warning sphere" objects would choke the draw-call limit of mobile hardware. By using GPU instancing, we tell the GPU to "draw this one sphere mesh 500 times at these 500 positions with these 500 colors." This technique, common in game development, is essential for industrial AR where thousands of data points might need simultaneous visualization.

Another critical aspect was the Data-to-Visual Mapping Logic. We discovered early in development that a 1:1 mapping (one number = one visual label) resulted in "visual clutter," a phenomenon termed by Tufte and adapted to AR contexts. If every sensor displays a floating text box, the user's view is obstructed by a cloud of text. We implemented a "Semantic Level of Detail" (S-LOD) system.

- Distance Phase: When the user is far from the equipment (more than 2 meters), the system aggregates data. Instead of showing pressures for "Pump A," "Pump B," and "Pump C," it shows a single status icon for "Pumping Station Alpha."
- Proximity Phase: As the user moves closer (physically walks toward the equipment), the aggregation dissolves, and individual sensor tags fade in.

● Inspection Phase: When the user is within 0.5 meters, detailed histograms appear floating next to the specific component.

This S-LOD system mimics how humans naturally interrogate objects: we assess the whole from a distance and scrutinize details up close. This required complex vector math within the Unity engine, constantly calculating the Euclidean distance between the camera's transform and every sensor node in the 3D space.

Furthermore, we had to address the Jitter Problem. Telemetry data often contains noise. A pressure sensor might read 100.1, then 99.9, then 100.2. If we mapped this directly to the scale of a graphical bar, the bar would vibrate distractingly. We implemented a Kalman Filter on the client side. This algorithm predicts the "true" state of the system by analyzing the statistical noise of the incoming stream over a sliding time window. The result was a smooth, readable visual transition, even when the raw sensor data was noisy.

Finally, the integration of Web-AR technologies was explored for lightweight deployment. While our primary experiment used a native app, we prototyped a version using Web-AR.Studio [11]. This allowed users to access the visualization via a browser without installing an app. However, we found that the browser-based access to the device's GPU and sensors was throttled compared to native code, resulting in higher latency. For critical safety operations, native applications remain the superior choice, though Web-AR shows promise for quick, ad-hoc inspections by non-specialized personnel.

The modeling of the assets themselves required precision. Using Blender [12], we created low-polygon "proxy" meshes of the real-world equipment. These proxies are invisible to the user but exist in the physics engine. When a user taps on a real-world pump on their screen, they are actually raycasting against this invisible proxy mesh. This highlights the importance of the alignment phase. We utilized image-recognition markers (QR codes) placed on the equipment. When the ARCore/ARKit camera recognizes the marker, it snaps the coordinate system of the virtual world to match the physical world. We achieved an alignment accuracy of ±2cm, which was sufficient for component-level identification.

4. RESULTS

The data collected from the 50 participants provided a clear picture of the strengths and weaknesses of AR in this context.

4.1 Reaction Time Analysis

The primary metric was the speed of anomaly detection.

● Control Group (2D Dashboard): The average time to acknowledge an anomaly was 12.4 seconds (SD = 3.2s).

● Experimental Group (AR Overlay): The average time was 9.4 seconds (SD = 2.1s).

This represents a 24.2% improvement in reaction speed for the AR group. The T-test analysis yielded a p-value of <0.01, indicating statistical significance. The reduced standard deviation in the AR group suggests that performance was more consistent across different participants. In the 2D group, some participants were very fast while others struggled to map the "Warning: Pump 4" text to the schematic, causing the higher variance.

4.2 Accuracy and Error Rates

Participants were asked to identify which specific component was failing.

● Control Group: 88% accuracy. Most errors involved misidentifying the component ID (e.g., confusing Pump 4 with Pump 5).

● Experimental Group: 96% accuracy. Since the alert was spatially attached to the object, "misidentification" was nearly impossible unless the tracking drifted.

4.3 User Experience and Cognitive Load

Using the NASA-TLX (Task Load Index) methodology, we assessed the perceived workload.

● Mental Demand: The AR group reported significantly lower mental demand (average 3.2/10) compared to the Control group (6.5/10). Participants described the AR experience as "obvious" and "intuitive," whereas the dashboard required "decoding."

● Physical Demand: Conversely, the AR group reported higher physical demand (5.8/10) compared to the Control group (1.2/10). Holding a tablet up for 20 minutes caused arm fatigue ("gorilla arm syndrome"). This is a critical finding for long-term implementation.

5. DISCUSSION

The results of this study strongly support the hypothesis that Augmented Reality can enhance telemetry interpretation, but the nuance lies in how and when it is applied.

5.1 The Power of Spatial Context

The 24% improvement in reaction time is attributed to the elimination of the "search" phase. In a traditional workflow, an alarm triggers a search: looking for the value, looking for the schematic, looking for the component. In AR, the alarm is the location. This aligns with the findings of Patel [15], who argues that real-time analytics benefits from reducing the abstraction layer. By placing the data on the object, we are effectively annotating reality.

5.2 Domain Implications

● Aerospace: As referenced by Cancro et al. [6], spacecraft controllers deal with thousands of parameters. An AR view inside the module could allow astronauts to look at a panel and "see" the coolant flow behind it, identifying a blockage instantly without consulting a laptop.

● Historical Preservation: Building on Rymarczyk's work [2], a conservationist could walk through a cathedral wearing AR glasses. Instead of holding a moisture meter against the wall every few feet, they could see a color-coded moisture map overlaying the masonry, revealing rising damp patterns that are invisible to the naked eye.

● Medical: Kaundal's work on video EEG telemetry [5] highlights the need for context. Future applications could involve projecting EEG data onto a 3D model of the patient's brain in real-time, allowing the neurologist to visualize the propagation of seizure activity spatially, rather than just temporally on a strip chart.

5.3 Cognitive Ergonomics and The "Gorilla Arm" Problem

(Expansion Section continues here)

While the cognitive benefits are clear, the ergonomic challenges discussed in our results section warrant a deeper theoretical analysis. This dichotomy—low cognitive load vs. high physical load—is the central tension in current AR deployment. The "Gorilla Arm" syndrome is not merely a comfort issue; it is an operational safety issue. If an operator's arm is tired, their hand shakes. If their hand shakes, the device's camera feed becomes unstable, which degrades the SLAM tracking quality, causing the digital overlay to jitter. This creates a feedback loop of degradation.

To mitigate this, we must look toward Mixed Reality (MR) Head-Mounted Displays (HMDs) rather than handheld tablets. Devices that move the display to the eyes and leave the hands free are the logical next step. However, HMDs introduce their own "vergence-accommodation conflict," where the eyes struggle to focus on a 3D object that is stereoscopically projected at 2 meters but optically displayed on a screen 2 centimeters from the retina. This can cause nausea and eye strain over prolonged periods.

We also observed an interesting psychological phenomenon we term "Tunnel Vision by Proxy." In the control group (2D screens), operators frequently scanned the entire dashboard, maintaining a high level of situational awareness of the whole system. In the AR group, operators tended to fixate on the specific component they were looking at through the screen. Because the field of view (FOV) of the tablet camera is limited (roughly 60 degrees), they effectively lost peripheral vision. If a warning light flashed on a component behind them, they missed it entirely until an auditory cue alerted them.

This suggests that AR should not replace global dashboards but augment them. A hybrid workflow is likely the optimal solution: a large, 2D "Situation Board" provides the global state (the "Macro" view), and the AR device is used for the "Micro" inspection of specific subsystems. This aligns with the "Overview first, zoom and filter, then details-on-demand" mantra of information visualization, updated for the age of spatial computing.

Furthermore, the integration of Historical Data into the AR view presents a massive opportunity. Rymarczyk's moisture analysis [2] relies on changes over time. A static AR overlay shows current status. A dynamic AR overlay could allow the user to scrub a time-slider. Imagine a structural engineer looking at a bridge support. By sliding a virtual dial, they can watch the crack propagation over the last five years replayed in high speed directly on the concrete surface. This temporal-spatial compression is impossible with 2D graphs (which have time but no space) or standard inspection (which has space but no time). We are effectively giving the operator 4D vision.

5.4 Technical Limitations

The current generation of mobile hardware, while impressive, still struggles with complex mesh occlusion. If a user walks behind a pipe that is not modeled in the proxy mesh, the virtual graphics will draw on top of the pipe, breaking the immersion. Accurate, real-time, dynamic occlusion (where the device understands the geometry of unmodeled objects instantly) requires significant machine learning processing power, currently pushing the limits of mobile NPUs (Neural Processing Units).

Battery life is another constraint. Running the camera, high-brightness screen, GPU, and Wi-Fi radio simultaneously drains a standard tablet battery in under 3 hours. For a 12-hour shift in a control room, this is unacceptable without hot-swappable power solutions.

6. CONCLUSION

This study set out to determine if Augmented Reality could serve as a viable medium for interpreting high-volume telemetry data. The results indicate that for tasks involving spatial identification and rapid anomaly detection, AR is superior to traditional 2D visualization. By leveraging tools like ARKit, ARCore, and Blender, we have demonstrated that effective visualization pipelines can be built with off-the-shelf software.

However, AR is not a panacea. It introduces physical fatigue and field-of-view limitations that must be managed through careful interface design and hardware selection. The future of telemetry is not entirely holographic, but it is certainly hybrid. We envision a future where the "screen" is no longer a rectangle on a desk, but a contextual layer woven into the fabric of the operational environment itself. As algorithms improve and hardware lightens, Immersive Analytics will likely become the standard for managing the complex systems that underpin our technological society.

REFERENCES

1. Read, A.J. T—Telemetry. In Encyclopedia of Marine Mammals, 2nd ed.; Perrin, W.F., Würsig, B., Thewissen, J., Eds.; Academic Press: London, UK, 2009; pp. 1153–1156.
2. Rymarczyk, T.; Adamkiewicz, P. Nondestructive Method to Determine Moisture Area In Historical Building. Inform. Autom. Pomiary Gospod. Ochr. Srodowiska 2017, 7, 68–71.
3. Woźniak, P.; Romanowski, A.; Yantaç, A.E.; Fjeld, M. Notes from the front lines: lessons learnt from designing for improving medical imaging data sharing. In Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, Helsinki Finland, 26–30 October 2014; pp. 381–390.
4. DeCelles, G.; Zemeckis, D. Chapter Seventeen—Acoustic and Radio Telemetry. In Stock Identification Methods, 2nd ed.; Cadrin, S.X., Kerr, L.A., Mariani, S., Eds.; Academic Press: San Diego, CA, USA, 2014; pp. 397–428.
5. Kaundal, A.; Hegde, V.; Khan, H.; Allroggen, H. Home video EEG telemetry. Pract. Neurol. 2021, 21, 212–215.
6. Cancro, G.; Turner, R.; Nguyen, L.; Li, A.; Sibol, D.; Gersh, J.; Piatko, C.; Montemayor, J.; McKerracher, P. An Interactive Visualization System for Analyzing Spacecraft Telemetry. In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 18 June 2007; pp. 1–9.
7. Guerra, J.C.; Stephan, C.; Pena, E.; Valenzuela, J.; Osorio, J. SystMon: A data visualization tool for the analysis of telemetry data. In Proceedings of the Observatory Operations: Strategies, Processes, and Systems VI, Edinburgh, UK, 1 July 2016; Peck, A.B., Seaman, R.L., Benn, C.R., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2016; Volume 9910, pp. 706–715.
8. Wang, C.; Ye, Z.; Wu, B.; Yin, H.; Cao, Q.; Zhu, J. The design of visualization telemetry system based on camera module of the commercial smartphone. In Proceedings of the Sensors, Systems, and Next-Generation Satellites XXI, Warsaw, Poland, 11–14 September 2017; Neeck, S.P., Bézy, J.L., Kimura, T., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2017; Volume 10423, pp. 348–355.
9. Mathur, M.; Brozovich, J.M.; Rausch, M.K. A Brief Note on Building Augmented Reality Models for Scientific Visualization. 2022.
10. AR - Augmented reality and its history [electronic resource] URL: https://habr.com/ru/post/419437/
11. Namiot, D.E.; Romanov, V.Y. 3D visualization of architecture and metrics software. Scientific visualization, 2018, volume 10, number 5, page 123 - 139, DOI: 10.26583/sv.10.5.08
12. Online platform for creating projects with augmented reality «Web-AR.Studio» [electronic resource] URL: https://web-ar.studio/ru
13. 3D modeling software «Blender» [electronic resource] URL: https://ww.blender.org/
14. 3D modeling software «3Ds max» [electronic resource] URL: https://ww.autodesk.com/products/3ds-max/
15. Online platform for accounting of analytics «Yandex metric» [electronic resource] URL: https://metrika.yandex.ru/
16. Dip Bharatbhai Patel 2025. Incorporating Augmented Reality into Data Visualization for Real-Time Analytics. Utilitas Mathematica . 122, 1 (May 2025), 3216–3230.
17. Apple Development Toolkit - ARKit [Electronic Resource] URL: https://developer.apple.com/augmented-reality/arkit/
18. Augmented Reality Game with Geolocation Recognition Algorithm - PokemonGO [Electronic Resource] URL: https://ww.pokemon.com/
19. Google software development tool - ARCore [electronic resource] URL: https://developers.google.com/analytics/