IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 3, Ser. 3 (May. – June. 2025), PP 17-23
www.iosrjournals.org

AlgoAR: Interactive Algorithm Visualization In Augmented Reality

Deepika Ajalkar[1], Nilamadhab Mishra[2], Anjali Shinde[3], Prarthana Kank[4], Shivanjali Bhilare[5]
(Dep. Of Data Science, G H Raisoni College Of Engineering And Management, Pune, India)

Abstract:
Background*: Understanding algorithms poses a significant challenge to students due to their abstract nature and the limitations of traditional teaching tools such as static diagrams and textbooks. Augmented Reality (AR) offers a transformative solution by enabling learners to visualize and interact with data in real time. AlgoAR, an AR-based learning platform, addresses this issue by immersing students in a 3D environment that demonstrates how algorithms function step-by-step. The tool aims to improve engagement, comprehension, and retention in algorithm education through immersive interaction and gamified elements.*
Materials and Methods*: The application was developed using Unity 3D and AR Foundation, integrating ARCore and ARKit for cross-platform support. Core algorithm logic, user interactions, and execution flow were implemented in C#. The system supports multiple algorithms including Bubble Sort, Merge Sort, DFS, BFS, and Dijkstra's Algorithm and features gesture-based manipulation and real-time visual updates. A usability evaluation and performance testing were conducted across multiple mobile devices to assess CPU usage, execution time, and responsiveness. Gamification features like quizzes, real-time feedback, and a leaderboard were also integrated to track learning performance.*
Results*: Experimental testing showed that AlgoAR maintained optimal performance on mid-range mobile devices, offering smooth rendering and accurate gesture-based manipulation. User feedback indicated a marked improvement in engagement and understanding when compared to traditional 2D algorithm visualization tools. Quiz-based evaluation demonstrated a notable improvement in conceptual clarity post-interaction.*
Conclusion: *AlgoAR successfully leverages augmented reality to bridge the gap between abstract algorithm theory and tangible understanding. By combining real-time visualization, interactive manipulation, and gamified learning, the tool fosters deeper comprehension and sustained engagement in computer science education.*
Key Word*: Augmented Reality AR; Algorithm Visualization; Interactive Learning; Gamified Education; Computer Science Education.*

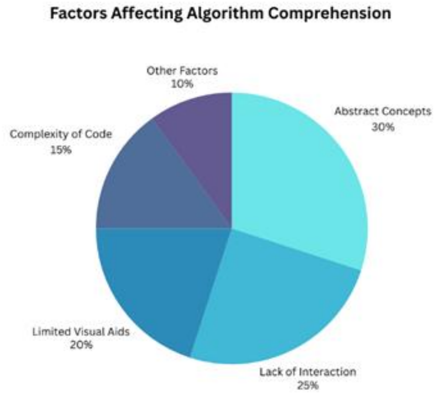
Date of Submission: 19-05-2025 Date of Acceptance: 29-05-2025

I. Introduction

Algorithms are fundamental to computer science education, yet students often find them difficult to grasp due to their abstract nature. Traditional learning methods rely heavily on textbooks, whiteboard illustrations, and 2D animations, which fail to provide a hands-on, interactive experience. Research suggests that interactive learning environments improve engagement and comprehension, making Augmented Reality (AR) a promising tool for algorithm visualization. AlgoAR is an interactive AR-based application that provides students with real-time, immersive experiences of algorithm execution. By placing 3D algorithm visualizations into a real-world environment, users can interact with data structures dynamically, observe step-by-step operations, and analyse performance metrics such as time complexity and execution steps. Additionally, gamification elements such as quizzes and challenges enhance learning motivation. The objective of this research is to develop and evaluate the impact of AlgoAR in improving students' understanding of algorithms through interactive AR-based learning experiences. The use of AR provides an interactive and immersive experience, making learning more engaging and effective, which aligns with existing studies that promote ICT tools for increased learner engagement.[1] [7]

Recent advancements in educational technology have introduced new opportunities to overcome these challenges. Augmented Reality (AR), a powerful medium that overlays digital content on the physical world, has emerged as a promising tool to enhance experimental learning. By integrating AR into the algorithm learning process, students can interact with digital data structures in a tangible, immersive way. This form of spatial computing supports kinesthetic and visual learning styles, enabling learners to physically observe how algorithms operate in real space, which promotes deeper understanding and engagement.[2][3]

Figure no 1: Factors Affecting Algorithm Comprehension

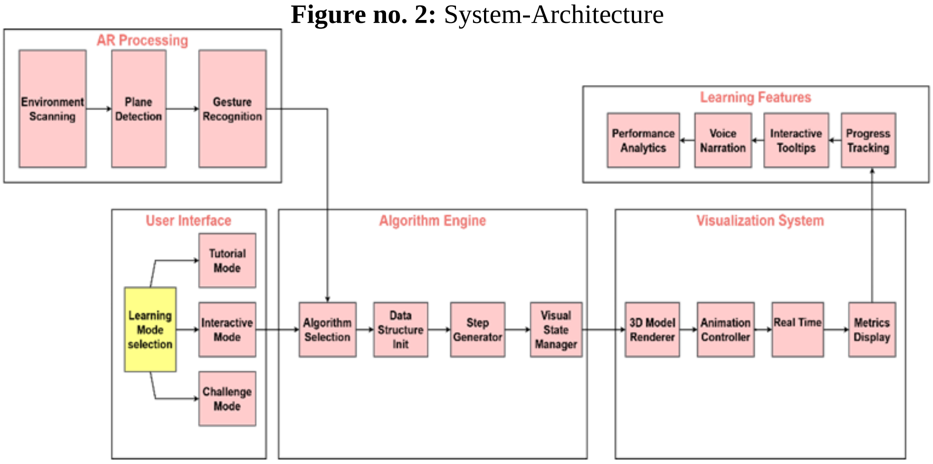


II. Material And Methods

System-Architecture

AlgoAR is designed with a robust system architecture that ensures seamless interaction between users and the application's augmented reality components. The system is composed of five key components, each playing a critical role in delivering an interactive and immersive learning experience.

Figure no. 2: System-Architecture



User Interface Layer: The UI layer is responsible for managing all user interactions, providing an intuitive and user-friendly interface that allows students to navigate through the platform effortlessly. Users can select algorithms, input custom datasets, and choose different visualization modes. The interface includes interactive elements such as buttons, sliders, dropdown menus, and real-time performance indicators, ensuring that users can dynamically control the learning process. Accessibility features such as voice-guided navigation and tooltips further enhance usability.

AR Processing Engine: The AR Processing Engine is the core component responsible for integrating the augmented reality functionalities. It uses plane detection algorithms to scan the user's surroundings and identify suitable surfaces for AR object placement.[4] The engine also supports markerless tracking, eliminating the need for predefined markers and allowing users to place the visualized algorithms in any real-world space. Gesture recognition capabilities enable users to interact with 3D models by rotating, resizing, and repositioning algorithmic structures, enhancing the hands-on experience. Additionally, real-time environmental adaptation ensures smooth AR rendering even in dynamic surroundings.

Algorithm Engine: This component is responsible for executing the algorithm logic and generating real-time visual updates. It supports a variety of fundamental sorting algorithms (Bubble Sort, Quick Sort, Merge Sort), graph traversal algorithms (BFS, DFS, Dijkstra's Algorithm), and data structure manipulations (binary trees, heaps, linked lists). The Algorithm Engine dynamically processes input values and executes each step of the algorithm, ensuring real-time visualization of comparisons, swaps, and insertions. Additionally, the engine calculates time complexity metrics and efficiency statistics, allowing users to analyze the performance of different algorithms side by side.

Visualization System: The Visualization System is responsible for rendering high-quality 3D models of data structures and animating each step of the algorithm execution. The system provides real-time visual cues, such as color-coded elements to indicate swaps, comparisons, and sorted portions of data. In addition, it includes dynamic overlays, explanatory tooltips, and pseudocode side-by-side representation, helping students correlate theoretical concepts with real-time execution. The visualization system also integrates camera angle adjustments, enabling users to zoom in, rotate, and view the algorithm processes from multiple perspectives.[2][5]

Learning Module: This module tracks user progress, quiz scores, and analytics on learning performance. It collects data on how students interact with the algorithms, providing personalized feedback based on engagement patterns and completion times. The module includes gamification elements such as quizzes, challenges, interactive problem-solving tasks, and progress tracking dashboards. These features enhance motivation by rewarding achievements, providing level-based progression, and encouraging users to improve their algorithm-solving skills through competition and self-assessment.

Technology Used

To develop a highly interactive and efficient AR-based learning platform, AlgoAR integrates several advanced technologies that ensure seamless functionality and high performance:

Development Platform–Unity 3D: Unity serves as the primary development environment due to its powerful 3D rendering capabilities, physics engine, and cross-platform compatibility. Unity enables the creation of interactive scenes, responsive UI elements, and real-time simulations, making it ideal for AR applications.

AR SDKs – AR Foundation (ARCore & ARKit): The AR capabilities of AlgoAR are powered by AR Foundation, a framework that provides cross-platform support for both Android (ARCore) and iOS (ARKit). The AR Foundation facilitates plane detection, motion tracking, and environmental understanding, allowing the smooth integration of AR elements into the real-world space. This ensures that the 3D algorithm models remain accurately positioned and responsive to user interactions.

Programming Language – C#: The application logic, including algorithm execution, UI interactivity, and AR component control, is implemented using C#. Unity's integration with C# ensures efficient memory management, optimized rendering performance, and real-time responsiveness. The C# scripts handle gesture inputs, animation triggers, and event-driven execution of the algorithms.

3D Modeling – Blender: Blender is used to design and optimize 3D models of data structures such as arrays, trees, and graphs. The models are designed to be lightweight and performance-optimized, ensuring smooth AR rendering on mobile devices. The Blender's powerful animation tools allow for seamless transitions between algorithmic steps, improving visual clarity.[6] [7]

Gamification and Learning Analytics: The platform incorporates Firebase Analytics and Machine Learning algorithms to track user performance, engagement levels, and learning progress. The data are used to personalize the user experience, suggest learning improvements, and adapt the content dynamically. Gamification elements are implemented using Unity's built-in physics and scoring system, allowing real-time leaderboard updates and interactive challenges.

Cloud Integration and Data Storage: The application can integrate with cloud-based storage solutions to allow students to save their progress, compare previous performance, and access personalized study materials. Cloud-based APIs enable collaborative learning features, where users can share algorithm simulations and discuss problem-solving strategies.

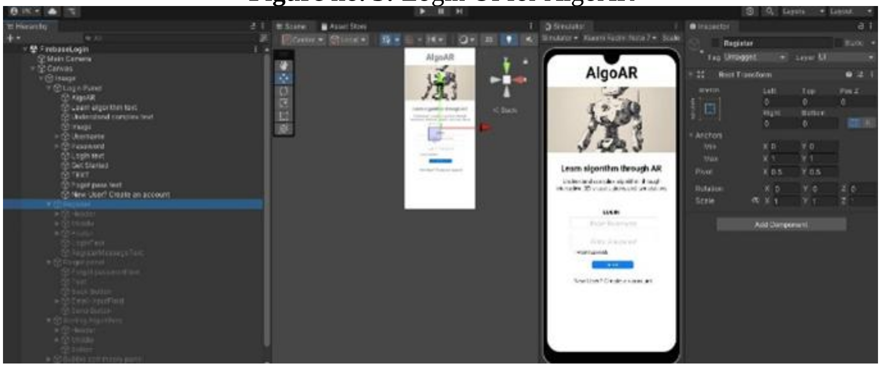
Implementation

Features:

AlgoAR is designed to provide an immersive and interactive learning experience by incorporating advanced Augmented Reality (AR) features, real-time execution, and gamification elements. The platform's core features
include:

Markerless AR Tracking: Unlike traditional AR applications that require predefined markers, AlgoAR uses markerless tracking technology powered by the AR Foundation, enabling users to place algorithm visualizations in any real-world environment. This allows for greater flexibility and eliminates the need for printed markers or predefined locations.

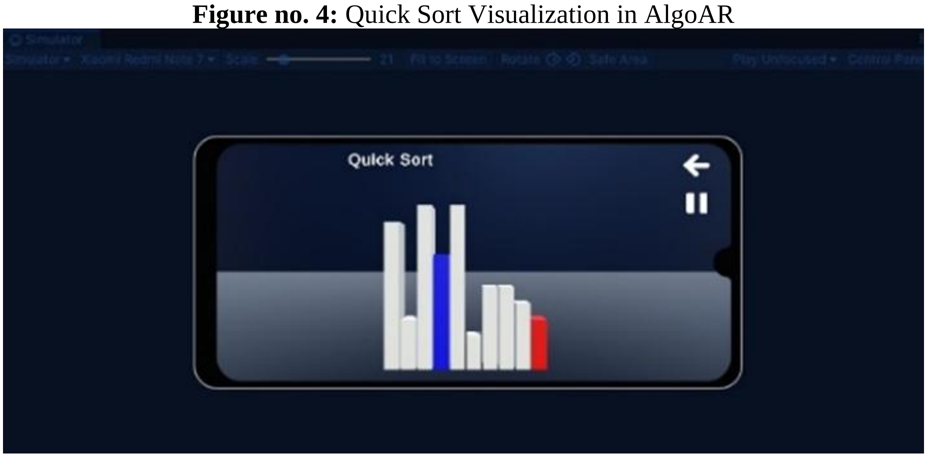
Figure no. 3: Login UI for AlgoAR



Gesture-Based Interactions: Users can rotate, resize, reposition, and manipulate algorithmic structures dynamically using intuitive touch and gesture controls. By pinching or dragging, students can modify the visualization to gain different perspectives, helping them understand the algorithmic processes from multiple viewpoints.

Step-by-Step Execution: The platform provides real-time, step-by-step breakdowns of the algorithm execution. Each step highlights key operations such as comparisons, swaps, recursive calls, and traversals with color-coded visual indicators. This interactive approach enhances comprehension by allowing users to pause, replay, or speed up the execution for deeper analysis.

Figure no. 4: Quick Sort Visualization in AlgoAR



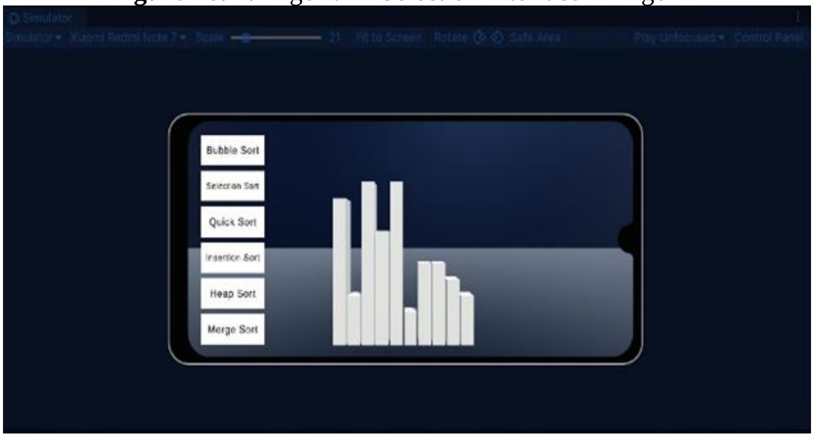
Gamification Elements: To enhance engagement, AlgoAR incorporates timed quizzes, interactive challenges, and leaderboard tracking. Students can test their knowledge by participating in quizzes that assess their understanding of different algorithms. The leaderboard system encourages competition, motivating students to refine their problem-solving skills and improve their performance.[7]

User Flow:

The user journey in AlgoAR is designed to be intuitive and seamless, ensuring a smooth learning experience from initialization to result analysis. The following steps outline the user flow:

Algorithm Selection: Users are presented with a menu of available algorithms, categorized by sorting, searching, and graph traversal techniques. They can select an algorithm based on their learning objective. Once an algorithm is selected, the 3D visual representation of the data structure appears in the AR space. The visualization includes interactive elements such as bars (for sorting), nodes (for graph traversal), and trees (for hierarchical structures).

Figure no. 1: Algorithm Selection Interface in AlgoAR



Real-Time Execution & Visualization: Once ready, users can trigger the execution of the selected algorithm. The system highlights each step dynamically, with color-coded indicators that distinguish comparisons, swaps, and recursive calls. The execution speed can be adjusted for better comprehension, enabling users to pause, rewind, or fast-forward the process.

Gamification & Learning Reinforcement: At key learning checkpoints, interactive quizzes and challenges appear, testing users' understanding of the algorithm. Users can earn points for correct answers, and their scores are tracked on a leaderboard.

Completion & Learning Progress Tracking: The learning module records user progress, including completed challenges, quiz scores, and time spent on different algorithms. Users receive personalized recommendations on which algorithms to revisit or explore next, ensuring continuous learning improvement.

Mathematical Model for AlgoAR

To formalize the functioning of AlgoAR, a mathematical model is developed to represent algorithm execution, visualization, user interaction, and performance evaluation within the augmented reality environment. This model ensures a structured approach to understanding how different system components interact dynamically.

Algorithm Execution Representation:

The execution of an algorithm in AlgoAR can be defined as a state transition system:

$$S_t = f\left(S_{\{t-1\}}, I_t\right)$$

Where:

- S_t represents the system state at time t, which includes the positions, colours, and structures of algorithm elements in AR space.
- S_{t-1} is the previous state before executing the next step.
- I_t represents user interactions, such as adjusting input size, selecting different algorithms, or modifying execution speed.
- f is the transformation function that dictates how the algorithm progresses from one state to another based on computational logic.

Each algorithm follows a well-defined transformation function f which varies based on its complexity (e.g., sorting algorithms like Bubble Sort follow element swaps, whereas graph algorithms like BFS follow node traversal sequences).

AR Rendering and Visualization Model:

The process of rendering an algorithm in augmented reality involves mapping algorithmic steps into a three-dimensional space. This transformation is given by:

$$V = R(A, U)$$

Where:

- V is the visual representation of the algorithm.
- A represents the algorithm's data structure (e.g., array for sorting, graph for traversal).
- U consists of user-defined parameters such as input size, execution speed, and viewpoint.

- R is the rendering function that maps logical algorithmic operations into 3D visual elements using Unity's AR Foundation.[7]

User Interaction And Gesture Recognition Model:

User interaction plays a key role in enhancing engagement with algorithm visualizations. Gesture-based interaction is modelled as a transformation matrix:

$$G = (R_x, R_y, R_z, S_x, S_y, S_z, T_x, T_y, T_z)$$

Where:

- G represents the transformation applied to the algorithm's visual elements.
- R_x, R_y, R_z define rotation along the X, Y, and Z axes.
- S_x, S_y, S_z represents scaling factors to adjust the visualization's size.
- T_x, T_y, T_z define translations, allowing users to reposition algorithm structures in AR space.

These transformations allow users to intuitively manipulate algorithm visualizations, enhancing interactivity and immersion.

Performance Evaluation Model:

To measure the effectiveness of AlgoAR, a learning performance model is established. Learning efficiency is defined as a function of engagement and accuracy in quizzes:

$$L = w_{1E} + w_{2A}$$

Where:

- L represents the overall learning effectiveness score.
- E represents engagement, which is measured by interaction duration, number of gestures performed, and frequency of feature utilization.
- Arepresents accuracy, determined through quiz scores, problem-solving exercises, and algorithm comprehension tests.
- $w_1, w_2$are weight parameters that determine the relative importance of engagement and accuracy.

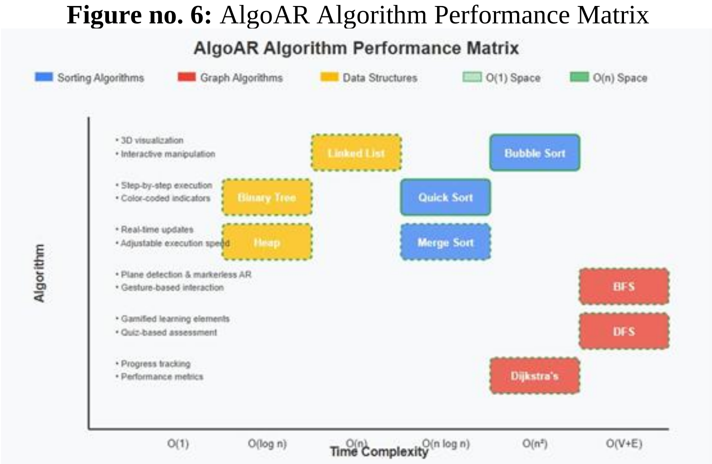
By analysing these performance metrics, AlgoAR can adaptively refine its instructional strategies to optimize user learning outcomes.

III. Result

To assess the effectiveness of **AlgoAR**, a series of experimental evaluations were conducted, focusing on performance, usability, and engagement. Performance metrics were analyzed by measuring execution time, CPU and memory usage, and rendering efficiency across different mobile devices.[8] [9] The algorithm execution time was recorded for varying input sizes to evaluate its efficiency, while resource utilization was monitored to ensure optimal performance on mid-range smartphones.

Figure 6 shows graphical representation of the algorithms and data structure used by AlgoAR according to their time and space complexity. It categorizes the items into three groups-Sorting Algorithms, Graph Algorithms, and Data Structures-using color-coded labels for better identification. The x-axis is for time complexity ranging from constant time ($O(1)$) to graph complexities ($O(V + E)$), and the y-axis is for the name of the algorithm or data structure.

Figure no. 6: AlgoAR Algorithm Performance Matrix



Sort algorithms such as Bubble Sort, Quick Sort, and Merge Sort are sorting and possess different complexities ranging from $O(n)$ to $O(n \log n)$, with respective space complexities ($O(1) or\ O(n)$). Graph algorithms such as BFS, DFS, and Dijkstra's Algorithm are placed on the extreme right since they possess greater time complexities and space requirements, suitable for traversal and pathfinding visualizations.

Data structures like Binary Tree, Linked List, and Heap fall under the lower-complexity category, in line with their application as fundamental visualization purposes. The matrix also shows fundamental elements of AlgoAR-such as 3D visualization, interactive manipulation, AR interaction through gesture, and learning elements through gamification-with computational complexity to create an effective and balanced learning process.

The performance matrix is employed to represent the broad range of algorithms employed, yet concurrently enable one to visualize the trade-offs between time and space consumption, thereby deriving algorithm insight via a graphical and interactive environment.[10]

IV. Discussion

Based on user feedback and performance evaluations, several enhancements were identified to improve the overall effectiveness of AlgoAR. Performance optimization emerged as a key area of improvement, focusing on refining the AR rendering pipeline to ensure smoother animations and reduced resource consumption. Implementing Level of Detail (LOD) models for complex visualizations and optimizing rendering through asynchronous execution techniques were proposed to enhance efficiency.[10] These improvements provide a seamless experience across mobile devices. These enhancements aim to create a more structured, immersive, and engaging learning journey for users of all backgrounds.

V. Conclusion

AlgoAR introduces an innovative approach to algorithm education by leveraging augmented reality to create an immersive and interactive learning experience. Traditional algorithm learning methods, such as textbooks and static visualizations, often fail to effectively convey the dynamic nature of algorithm execution, making it difficult for students to grasp complex computational concepts. AlgoAR addresses this challenge by providing real-time, step-by-step visualizations that allow users to observe how algorithms function in a three-dimensional space. With features such as **gesture-based interactions, comparative analysis, and gamification elements**, the platform significantly enhances user engagement and comprehension. By enabling learners to interact with algorithms dynamically—modifying input sizes, rotating visualized structures, and tracking real-time execution metrics—AlgoAR fosters an intuitive and hands-on learning experience.

References

[1]. Chen Chen, Y. T. (2018). The Exploration Of Augmented Reality Technology Applied In Colleges' Experimental Teaching . 9th International Conference On Information Technology In Medicine And Education.

[2]. Chhaya Nayak, D. A. (2023). Machine Learning Thyroid Model For Prediction System. International Conference On Device Intelligence, Computing And Communication Technologies.

[3]. Esron Rikardo Nainggolan, H. H. (2018). The Implementation Of Augmented Reality As Learning Media In Introducing Animals For Early Childhood Education. The 6th International Conference On Cyber And IT Service Management (CITSM 2018) . Inna Parapat Hotel – Medan.

[4]. Firoz Khan, J. A. (2020). Hybrid Reality Based Education Expansion System For Intelligent Learning. Seventh International Conference On Information Trends (ITT).

[5]. Guang-Jin Chen, R.-S. C. (2018). A Case Study Of The Development Of Using Augmented Reality In Teaching Nature And Life Technology To A Junior High School Students In Southern Taiwan . 1st International Cognitive Cities Conference (IC3).

[6]. Hunhui Na, K. B. (2023). Investigating The Impact Of AR Technologies On Geometric Learning In Primary School: A Comparison Between Marker-Based And Markerless AR. British Journal Of Educational Technology, 54(1), 174–190.

[7]. Mukasheva, M. K. (2023). Visualization Of Sorting Algorithms In The Virtual Reality Environment. Frontiers In Education, 8(1195200).

[8]. Nilamadhab Mishra, S. M. (2022). A Novel Intrusion Detection System Based On Random Oversampling And Deep Neural Network. Indian Journal Of Computer Science And Engineering (IJCSE), 13(6), 1924-1936.

[9]. Uriarte-Portillo, A. Z.-C.-E.-B.-M. (2023). Intelligent Augmented Reality For Learning Geometry. Information. ACM Digital Library, 28(1).

[10]. Yang Kuang, X. B. (2019). The Feasibility Study Of Augmented Reality Technology In Early Childhood Education. The 14th International Conference On Computer Science & Education (ICCSE 2019). Toronto, Canada.