Visualizing Sorting Algorithms with ReactJS: Merge Sort and Quick Sort Demystified

Dr. Neha Gupta[1], Prateet Mishra[2], Khyati Patel[3], Palak Gandhi [4], Mustafa Panjiwala[5],

[1,2,3,4,5]School of CSIT, Symbiosis University of Applied Sciences, Indore, India

Abstract

This study examined the advantages of an animated sorting algorithm for both teaching and learning objectives. Here, to visualize merge sort and quick sort, a web-based application was constructed. The UI will contain options to select one of the sorting algorithms which were implemented and several items or elements in the data array, control buttons to start, pause, and navigate to previous or next steps along with an option for sorting speed and color mode. The data array of the selected size will be filled in with randomly generated unique values. The data set is represented as a vertical bar with the height of their respective values. After the sorting is started, the stepwise arrangement of data in ascending order based on their value/height will be visualized in the UI.

Keywords and Phrases: Sorting Algorithm, Data Array, Information Technology (IT), User Interface (UI), JavaScript (JS), Personal Computer (PC), Algorithm Visualizer (AV).

1. Introduction

Information is the IT business' most sought-after product right now. Data management and meaningful data organization are just as important as data collection and storage. Data organization helps us to arrange the data in order so that we can read and work on that easily. Dealing with unorganized or raw data is cumbersome and takes huge amount of time to extract knowledge from it.

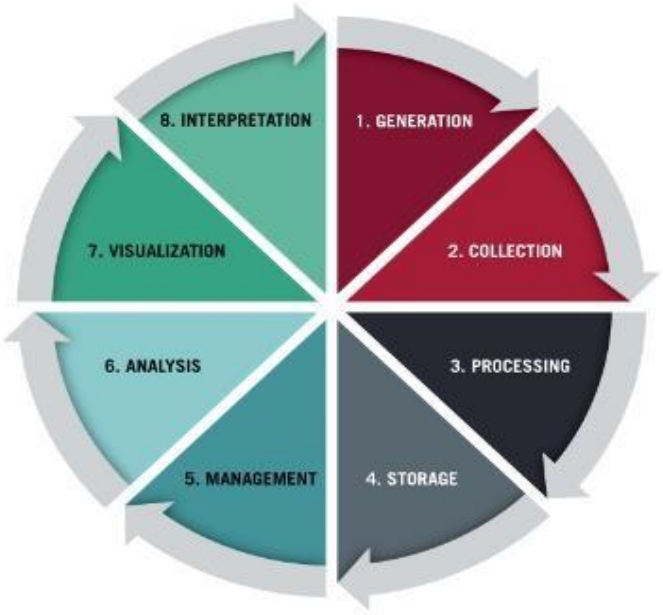

***Figure* 1.1:** 8 steps in the data life cycle.

This data life cycle helps us to understand the stages through which data goes, in order to be interpreted judiciously. The data life cycle is commonly depicted as a cycle since the knowledge acquired and insights gained from a previous data project typically influence the next one. Thus, the concluding stage of the process provides input to the initial step. Data organization is also a part of this cycle which encloses collection and processing stages of the lifecycle. Below is the figure of the same.

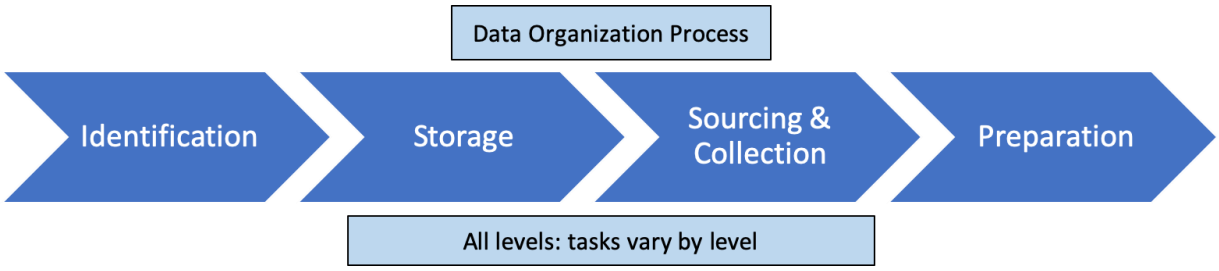

***Figure* 1.2:** Data organization Process.

Organizing of data consist of sorting of data, which provides a proper structure to the data making it comparable to another set of data. For example, if we open file explorer on your PC, you may see files which are sorted according to different categories. This is because searching in sorted data is more efficient and efficacious than in not sorted ones. As we faced the problem in understanding the working of sorting algorithms, somewhere we understood that the visual representation is the vital part of the studying process [4]. Now, let's imagine we have a database of millions of customer entries and we need to sort them by their first name. In order to do this, we could divide the database into smaller packets, sort those packets separately and then merge those packers again into a sorted database. This technique is quite simple to explain to someone in conversation. But more advanced algorithms, like quick sort, which requires data point to be moved around a pivot point, requires strenuous efforts to understand using text alone. In order to appeal comprehensive individuals, we made our animations through various technology media, in a web based format. Instead of installing some additional software or setup some tools, it helps users to detach from this source of anxiety. It is made using React JS.

2. Literature Review

The main objective of sorting algorithm visualizers is to provide a visual representation of the sorting process, thereby assisting users in comprehending how various sorting algorithms operate. This review aims to examine the effectiveness of sorting algorithm visualizers in enhancing understanding and knowledge retention by analyzing the diverse studies and research conducted in this area.

As we all know, visual representation of a difficult task can alleviate our stress so that we can understand it in a much better way. Similarly with sorting algorithms, it would be much easier for students to understand the functioning of algorithms through visualization [5]. This can be done by constructing an application to understand sorting as is the key component of Data Structures and Algorithms and it gets complicated and for the students to understand the whole algorithm and code for the first time. This project helps beginners to visualize the algorithms so that our brain can understand faster and retain better [6]. In 2016, the concept of interactive algorithm visualization for hybrid mobile application (INAVOHMA) was proposed to help an IT student learn the subject of Data Structure and Algorithm (DSA). [10]. In 2017, a group of people constructed DAVE, a web-based dynamic algorithm visualization environment designed to support secondary education students' learning about basic algorithms. DAVE makes it easy for students to experiment with spreadsheet algorithms by allowing them to modify both code and data. The presentation of preliminary results of the evaluation study provided evidence of the usability of the system and its potential to support students' development of effective mental models of core set algorithms [7]. In addition, new features can be integrated

in order to expedite the process of learning like a step-by-step visualization of the sorting process [8] where the user can learn how data is being sorted at each step and how we are getting the sorted data as a final result. This will motivate students to learn from such type of AV's more and will aid them its practical implications and educational benefits too. Also a comparison can be set between two different algorithms on their performances, utility, etc. This way students can implement a new customizable algorithm visualization tool which will help them to understand any algorithm while keeping them motivated and interested in the learning process [19]. Even professors can use different AV's to teach people in classrooms. This will aid professor to make students understand sorting algorithms and its process in an easier way while students can learn and retain knowledge faster and for a longer period of time through its visualization [2].

Although sometimes it is not necessary that students will really understand everything through just visualizing it. By just seeing the representation of the process, there is an uncertainty if students are really understanding it or not as they are not actively involved in the entire process, except some. Keeping this in mind, gamification of AV's started taking place. The 2022 study used the Sorting Algorithm Serious Game to allow students to learn four sorting algorithms: bubble sort, selection sort, insertion sort, and quick sort. Students completed self-guided lecture material in a serious game, then refreshed their learning with a visualizer, and finally reinforced their learning with a serious sorting game. An experimental group challenged with sorting algorithm games outperforms a control group that learns without a serious game. Game-based learning provides students with a positive learning experience that can improve learning performance. Combined with technology such as VR headsets as a future innovation, this would be a niche factor to create immersive learning that engages students and enhances their learning in a virtual environment. The idea would not only help students and learners better understand the concepts of algorithms, but also provide an innovative way for teachers and trainers and enhance their learning in a virtual environment [15]. Similarly, to change the perception of this subject from complex and hard to grasp, to interesting and fun. In 2021, a group of people implemented a work which aims to engage the students by providing self-paced hands-on experience, fun filled games through mazes and patterns and interactive, perceivable visualizations for their better concept understanding of various algorithms. The work presently focuses on path-finding, sorting and CPU scheduling algorithms as these are the most widely taught and used algorithms in the computer science domain. The idea would not only help the students and learners get a better hold on the concepts of algorithms but also provide an innovative way for teachers and educators to portray their ideas more clearly and interactively through to the students [18]. Algorithmic visualization, an attempt to solve a problem, has recently grown into a playful visualization that should be able to engage learners longer and more intensely. However, integrating algorithm visualization, game elements and instructional design is not a trivial task as it requires careful planning. Therefore, a conceptual model of how algorithmic learning instructions, algorithm visualization and gamification improve learning outcomes was developed. While curriculum is about developing the best learning strategy, visualization algorithms work as a cognitive support provider and gamification works for engagement. Both cognitive support and engagement moderate study instructions related to improving learning outcomes. During the design process, the principles and elements of the three domains must be considered to create an artifact that can fulfill the basic functions of each domain. After that, an artifact functioning as an implementation of the model was built, which confirms whether the idea of integrating algorithm visualization and play into instruction algorithms is sufficiently capable of improving learning outcomes. Based on the learning results of the respondents, it was found that the artifact can significantly improve the procedures of the students, which indicates their increased ability to solve the sorting algorithm tasks. [16].

Moreover, AV's are helping especially abled, here visually impaired students, to learn about different sorting algorithms. This can be achieved by integrating specific sounds with the sorting algorithm program. As the program accesses a value being sorted, the program plays a tone whose pitch is scaled to that value's magnitude. In the resulting sonifications, one can (in real time) hear the behavioral differences of the different sorting algorithms as they run, and directly experience how fast (or slow) the algorithms sort the same sequence, compared to one another. This will aid students with visual impairments to study about algorithms [11].

In order to believe the efficacy of the AV's different experiments were conducted and results were established

by comparing the novel visualization techniques with the traditional ways of teaching algorithms and data structures [20]. Also, a systematic meta-study of 24 experimental studies presented a way to better understand the effectiveness of Algorithm Visualizer (AV) technology. They pursued two separate analyses: an analysis of independent variables, in which we tie each study to a particular guiding learning theory in an attempt to determine which guiding theory has had the most predictive success; and an analysis of dependent variables, which enables us to determine which measurement techniques have been most sensitive to the learning benefits of AV technology [12].

3. Scope

A sorting algorithms visualizer's scope would normally include the range of algorithms that it can implement and visualize, as well as the level of information and customization that users have access to [14]. Here are some key factors that will influence the visualizer's scope:

A variety of sorting algorithms should be implemented and visualized by the visualizer, here quick sort and merge sort.

A) Detail level:

The visualizer should give multiple degrees of detail to meet the needs of different users. Some viewers may choose to view the full sorting process step by step, but others may prefer to concentrate on specific aspects of the algorithm's behavior, such as the comparison and swap operations. To achieve these requirements, the visualizer should provide options for adjusting the amount of information.

B) Customization:

The visualizer should allow users to tweak the sorting algorithms to some extent. Users should be able to add their own data sets to be sorted and choose the order in which the algorithms sort the data [19] For example, users should also be able to tailor the pace of the sorting algorithm and the visual display to their preferences.

C) Platforms and languages:

To guarantee most extreme client openness, the visualizer ought to be accessible on a scope of stages and programming dialects. The visualizer, for instance, ought to be accessible as a web application that is compatible with a wide range of operating systems, including Windows, Mac, and Linux, as well as the most recent versions of web browsers. The visualizer ought to likewise be made in notable programming dialects like JavaScript, Python, and Java, which are widely utilized in the programming local area.

A sorting algorithms visualizer's overall scope would include offering a wide selection of sorting algorithms, multiple degrees of information to suit different users, customization possibilities, and accessibility across different platforms and programming languages.

4. Proposed Methodology

The first thing that needs to be done is to look into the sorting algorithm visualizers and other tools that are related and find any potential market gaps. The development of a distinctive, useful, and valuable visualizer for users would benefit from this information. The visualizer's specific features and functionalities should also be defined and a detailed development plan should be developed. We are using ReactJS to build the project's frontend because it has cutting-edge functionality that allows UI interactions to communicate with JavaScript-driven pages in record time, saving us time by avoiding full page reloads. The code also resembles a Model-View-Controller architecture where sorting algorithms works as “Model”, JavaScript as “Controller” and “HTML User Interface” as a view. Below is the diagram showing the relationship between

the three of them.

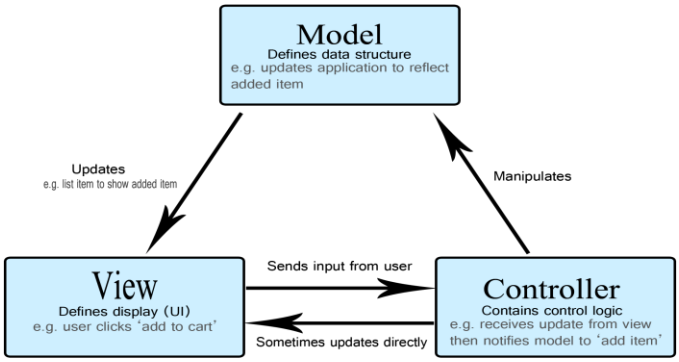

***Figure* 4.1:** MVC Architecture Diagram.

The development of the visualizer's user interface and data model would be the next step. This would entail creating a data model to support the sorting algorithms and designing the visualizer's layout and interactions. The data model ought to be capable of storing data sets supplied by the user, sorting the data using a variety of algorithms, and displaying the sorted results in a way that is simple to comprehend. When the UI and information model have been created, the subsequent stage is execute a scope of arranging calculations in the application and its interactivity with the user [17]. This would involve fostering the fitting activities to delineate the arranging system and composing code to deal with every calculation. Clients ought to likewise have the option to redo the arranging system and sort various kinds of information utilizing the execution.

After the visualizer has been developed, numerous tests would be required to identify any issues or bugs. This would entail running the visualizer with a variety of data sets and algorithms and testing the user interface and experience. Any issues that are found ought to be settled instantly to guarantee that the visualizer is easy to understand and proceeds true to form. It is possible to put the visualizer into production after it has been thoroughly tested and all issues have been resolved. To keep the visualizer up to date and compatible with changes to the underlying technologies, regular maintenance should be performed. The visualizer might should be refreshed to help new arranging calculations or information types, and the UI or execution issues may likewise should be tended to.

5. Design and Implementation

The design and structure of the application has remain unchanged even if the back-end code was changed midway of the implementation. The GUI of the application consists of 6 components with their own features. The panel is a space where the sorting algorithms, here merge sort and quick sort, are visualized with the help of bar graphs serving as individual data [1]. The navigation panel will consist of sorting algorithm selections and their comparison. This type of visualization helps users to select algorithms of their choice and to observe how that algorithm functions. Before starting with the animation, the user requires to select an algorithm. Below the panel, where the sorting algorithm is visualized, the user will get three buttons: sort, reset and next.

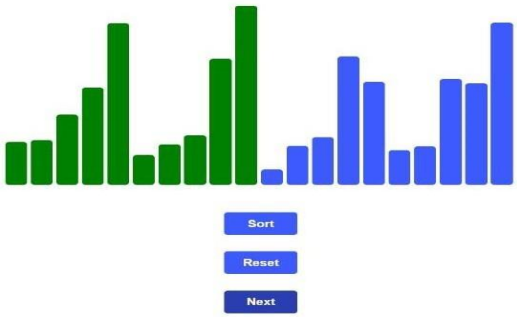

***Figure* 6.1:** Shows one part is sorted and other is yet to be sorted.

If a user wants to sort the data in one go, here represented as bar graph, can use the sort button shown in the figure above. The sorting will begin after clicking on the sort button. The visualization consists of 3 colors of bars, blue, red and green which are depicting unsorted data, comparing data, and sorted data respectively.

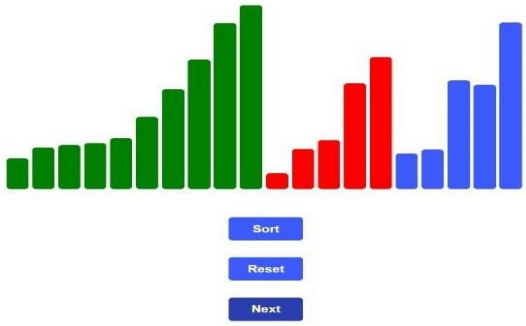

***Figure* 6.2:** Shows the second part started to get sorted.

If a user wants to reset the sorted data to random data, Reset buttons comes into play. On clicking the reset button, it will arrange the data in random order. Now the user can again use the sorting algorithm of their choice, here merge sort or quick sort, on that data.

***Figure* 6.3:** Shows randomized dataset after clicking on reset option.

In order to understand each step in sorting process, a user can use “NEXT” button to sort the given data one by one. After the sorting process completes, the panel will show sorted data in green colored bar graph.

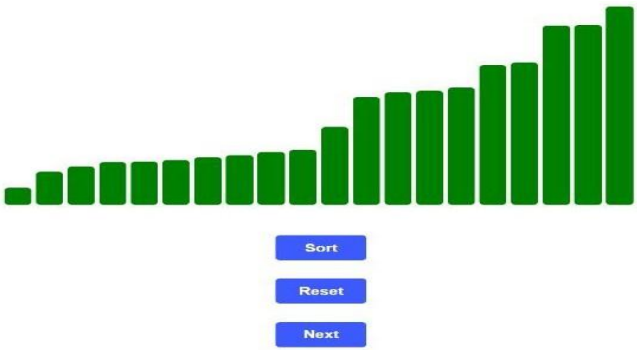

***Figure* 6.4:** Shows sorted dataset after the completion of sorting process.

6. Unit Testing

Unit A sorting algorithm visualizer's functionality as well as its user interface components are both tested during unit testing.

To use the next and previous buttons to move through the phases of sorting one at a time, as well as the sort button that animates the sorting in one go, on the website that represents Merge Sort using bars:

i. Open the website in your web browser by entering the URL in the address bar and pressing Enter.

ii. As soon as the page loads, you should see a bar-interface that represents and unsorted array.

iii. Look for a "Sort" button on the page. Clicking this button should trigger an animation that sorts the array using the Merge Sort algorithm in one go. Note that the bars representing the array will move and rearrange themselves during this animation.

iv. Look for “NEXT” and “PREVIOUS” buttons on the page if you would want to proceed through the Merge Sort algorithm’s steps one at a time. The “NEXT” button should show the following stage of the Merge Sort Algorithm, and the “PREVIOUS” button should show the preceding stage.

v. The array –representing bars will change to reflect the current phase in the Merge Sort algorithm as you click the “NEXT” and “PREVIOUS” button. As you processes through the processes and read the brief descriptions for each step for a better understanding, you might notice that the bars themselves move and rearrange themselves.

vi. Press the “NEXT” or “PREVIOUS” button repeatedly until the array is completely sorted. Keep in mind that the “NEXT” button will become inactive once you reach the end of the steps, signifying that there are no more steps to display.

vii. After the array has been completely sorted, you can take a minute to admire the Merge Sort algorithm’s visual depiction. You might also want to look at any extra options or features that the website provides, such the capacity to modify the dimensions or contents of the array being sorted.

7. Conclusion

This research paper is a web based sorting algorithm visualizer application that helps in sorting the given array by applying different sorting methods, here consolidated sort and speedy sort. Arranging calculation visualizer are integral assets that assist us with acquiring bits of knowledge about how different arranging calculations work by giving a visual portrayal of the whole cycle With the assistance of this we can acquire a more profound comprehension of how they work and how various they are concerning productivity and speed. Besides, they can assist us with streamlining our code and work on its exhibition by distinguishing likely issues and inefficacies. By exploring different avenues regarding different information values we can acquire bits of knowledge to the qualities and shortcomings of the calculation which will assist us with knowing which sort of calculation is expected for our undertaking.

References

[1] Shubham Nath, Jatin Gupta, Abhinav Gupta, Teena Verma, "SORTING ALGORITHM VISUALIZER", International Research Journal of Modernization in Engineering Technology and Science, Vol-03, Issue-07, July 2021

[2] Neelu yadav, Amisha Tripathi, "SORTING ALGORITHM VISUALIZER", International Journal of Advance Research and Innovative Ideas in Education, Vol-08, Issue-03, ISSN NO. 2395-4396, 2022.

[3] Wee Han Lim, Yiyu Cai, Dezhong Yao, Qi Cao, "Visualize and Learn Sorting Algorithms in Data Structure Subject in a Game-based Learning", IEEE International Symposium on mixed and augmented reality adjunct, 17-21 October, 2022.

[4] K.T.S Kasthuriarachchi, "AN INTERACTIVE VISUALIZATION TOOL FOR LEARNING SORTING ALGORITHMS", 7th International Conference on Education and New Learning Technologies, Pages 2678-2682, 6-8 July, 2015, Barcelona, Spain.

[5] Aditya Thakkar, Kavita, Sonali Dash, Sanjeev Kumar Joshi, "Sorting Algorithm visualizer", International Conference on cyber resilience, 6-7 October, 2022, Dubai, United Arab Emirates.

[6] Naziya Sheikh ,Ishna Sheikh , Anam Kusar Khan , Shail Rahangdale , Priyanka Gaikwad , Ashwini Pathare , Prof Kamlesh Kelwade, "ENHANCED APP BASED SORTING ALGORITHM VISUALIZER", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 11, Issue 3, March 2022.

[7] Euripides Vrachnos, Athanassios Jimoyiannis, "Design and evaluation of a web-based dynamic algorithm
Visualization environment for novices", 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, DSAI 2013.

[8] Bremananth R, Radhika.V and Thenmozhi.S, "Visualization of Searching and Sorting Algorithms", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol: 3, No: 3, 2009.

[9] Prof. Asha P , Gurpreet Kaur , Anmolpreet S , Shashank K , Akshay Kumar, "ALGORITHM VISUALIZER", IRE Journals, Vol: 6, Issue: 1, ISSN: 2456-8880, July 2022.

[10] Ahmad Affandi Supli, Norshuhada Shiratuddin, Syamsul Bahrin Zaibon, "Critical Analysis on Algorithm Visualization Study", International Journal of Computer Applications 150(11):18-22, September 2016.

[11] Joel C. Adams, Bryce D. Allen, Bryce C. Fowler, Mark C. Wissink, Joshua J. Wright, "The Sounds of Sorting Algorithms: Sonification as a Pedagogical Tool", Proceedings of the 53rd ACM Technical Symposium on Computer Science Education, Vol: , pages 189-195, February 2022.

[12] Christopher D. Hundhausen, Sarah A. Douglas, John T. Stasko, "A Meta-Study of Algorithm Visualization Effectiveness", Journal of Visual Languages and Computing, Vol: 13, pages 259-290, June 2002.

[13] Prathamesh Karanjkar, Viraj Patil, Shrikant Gupta, Prof. Reena Deshmukh, "Cognitive Visualization", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol: 10, Issue: 4, ISSN: 2321-9653, April 2022.

[14] Georgi Tuparov, Daniela Tuparova, Vladimir Jordanov, "Teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course", 5th World Conference on Educational Sciences - WCES 2013.
[15] Wee Han Lim, Yiyu Cai, Dezhong Yao, Qi Cao, "Visualize and Learn Sorting Algorithms in Data Structure Subject in a Game-based Learning", IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), 17-21 October 2022, Singapore.
[16] Alfa Yohannis, Yulius Prabowo, "Sort Attack: Visualization and Gamification of Sorting Algorithm Learning", 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games), 16-18 September 2015, Skovde, Sweden.
[17] Pálma Rozalia Osztián, Zoltán Kátai, Erika Osztián, "Algorithm Visualization Environments: Degree of interactivity as an influence on student-learning", IEEE Frontiers in Education Conference (FIE), 21-24 October 2020, Uppsala, Sweden.
[18] N. Gupta "Neural Network using RGB Gray Scale watermarking and subsequent Union of RGB planes Embedding in a Digital Image.", Turkish Journal of Computer and Mathematics Education, Vol 10 Issue 3, pp.987-994 June 2019
[19] Barnini Goswami, Anushka Dhar, Akash Gupta, Antriksh Gupta, "Algorithm Visualizer: Its features and working", IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 11-13 November 2021, Dehradun, India.
[20] Y. Prasad, N. Gupta "Implementation of Machine Learning Based Google Teachable Machine in Early Childhood Education", International Journal of Early Childhood Special Education (INT-JECSE), Vol 14 Issue 3 (2022), pp.4132–4139.
[21] Md. Abdul Matin, Sha Soultan Md, Oliullah, Md. Masbaul Alam Polash, "Implementation of a Customizable Algorithm Visualization Tool for E-Learning", Proceedings of the 2nd International Conference on Education and E-Learning, pages 32-36, November 2018.
[22] N. Gupta "Embedding color watermark in a digital image by adjusting DCT coefficients through Back Propagation Neural Network using RGB Gray Scale watermarking and subsequent Union of RGB planes.", International Journal of Test & Management, Vol 83 Issue 3 (2020), pp.375-380.
[23] Slavomír Simoňák, "Algorithm visualizations as a way of increasing the quality in computer science education", IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 21-23 January 2016, Herlany, Slovakia.