Sparking Light Publisher

Sparklinglight Transactions on Artificial Intelligence and Quantum Computing

journal homepage: https://sparklinglightpublisher.com/

Systematic Literature Review on Parallel Sorting and Searching Algorithms for Graph Analytics Problems

Shajil Kumar P A[a#], R. Srinivasa Rao Kunte[b], Pallavi Vaidya[c]

[a#]*Research Scholar, Institute of Computer Science and Information Science, Srinivas University, Mangalore, India*
Assistant Professor, Vidyalankar School of Information Technology, Mumbai, Maharashtra, India
[b]*Research Professor, Institute of Computer Science and Information Science, Srinivas University, Mangalore, India*
[c]*Professor, Seneca Polytechnic, North York, Ontario, Canada*

Abstract

This research paper aims to systematically review the literature published recently on the topic of parallel processing algorithms for sorting and searching methods. The paper emphasizes finding the major gaps in the existing methods of using parallel sorting and searching methods for graph analytics problems and their use in scalable data science. Many peer-reviewed research papers and articles from international and national journals, conferences and other sources are searched using appropriate keywords and they are analyzed to understand the concept, current trend in the area and to find the research gaps, by employing a systematic review method of literature. The outcomes of this literature review have resulted in the need to undertake a research work on "Design and Implementation of Parallel Sorting and Searching Algorithms for Graph Analytics Problems in Scalable Data Science". Also, we have identified the required research objectives and its plan of execution for the research proposal. Systematic literature review, Topic of research proposal along with research objectives, Listing of SWOT analysis of the research agendas, Listing of ABCD analysis of the research proposal.

Keywords: Parallel Sorting, Parallel Searching, Graph Analytics, Scalable Data Science.

1. Introduction

In computer science, sorting and searching algorithms are considered as the basic algorithmic procedures that are used to organize the data in a specific order and to fetch the information from different sets of data with minimum time. Sorting is the method of organizing the set of data items in a specific order like numerically ascending or descending way or in alphabetical order. Commonly used sorting techniques are bubble sort, selection sort, insertion sort, merge sort, quick and heap sort. Searching is the method of discovering a specific element from a set of data items. Commonly used searching techniques are linear search, depth-first search, breadth-first search, binary search, hashing etc.

Sorting and searching methods are conventionally developed with sequential procedures where execution of the instructions takes place in serial order. The efficiency of these algorithms can be improved by implementing them to execute in parallel. Parallel sorting and searching algorithms are used to perform the operations designed for specific computing operations on large data sets which are generally known as big data.

Email address of Authors: shajilkumar.pa@gmail.com, kuntesrk@gmail.com,
pallavi.vaidya@senecapolytechnic.ca

Please cite this article as: Shajil Kumar P A, R. Srinivasa Rao Kunte, Pallavi Vaidya (2023). Systematic Literature Review on Parallel Sorting and Searching Algorithms for Graph Analytics Problems. Sparklinglight Transactions on Artificial Intelligence and Quantum Computing, 3(1), 1-15. **ISSN (Online):2583-0732.** Received Date: 2023/04/25, Reviewed Date: 2023/05/20, Published Date: 2023/06/30.

The procedure of implementing parallel sorting and searching involves the method of dividing the problem among multiple processors or threads of execution that work concurrently to enhance the overall performance of the procedure and to reduce the execution time requirement of the procedure.

Graph analytics are methods to analyze the relationships between entities with the graph data structure to get meaningful ideas and patterns from the complex data sets. Graph data structures consist of a set of nodes or vertices and edges connecting the nodes. There are numerous procedures in graph analytics like graph connectivity, path analysis, community detection, clustering, graph traversal, node centrality etc. There are numerous applications of graph analytics in today's world in various domains of studies including science, management, economics, and finance. Some of the areas where graph analytics are used include social medial network analysis, fraud detection, biological network analysis, transportation problems and knowledge graphs.

Parallel sorting and searching procedures can be used to enhance the efficiency of the graph analytics problems. Most of the graph analytics procedures require traversing a set of nodes or edges, searching a particular node or edge, or retrieving the edge depending on the order of the edge weight. These operations become complex and time consuming when the graphs are larger in scale. To overcome the execution delays in the graph analytics problems parallel searching and sorting techniques can be used when the graphs become scalable.

2. Objectives of Review Paper

Our literature review has the following objectives:

1) To synthesize the current literature available on the topics of Parallel Sorting Techniques, Parallel Searching Algorithms, Graph Analytics Problem, and Scalable Data Science.
2) To identify the research gap and search for opportunities to address the research gap.
3) To develop a research agenda and bridge the gap between the present status and ideal status.
4) To evaluate the research agenda using SWOT analysis
5) To construct a research proposal and perform an ABCD analysis on the same.

3. Methodology

Our literature survey is conducted by examining numerous electronic databases of research repositories like Google Scholar, Research Gate, IEEE Xplore, ShodhGanga, Directory of Open Access Journals (DOAJ), Scopus and Web of Science. Various manuscripts pertained to the study like journal articles, conference-papers, books and chapters were studied as part of this research work. Articles published from 2010 to 2023 were accessed through Google Scholar and many are referenced in this research work as part of the literature review. Research work including research papers, conceptual papers, review papers and articles have been accessed with the keywords of search Parallel Sorting, Parallel Searching, Graph Analytics and Scalable Data Science.

Text based database exploration has been done based on the keywords of search to gather the relevant data from the research articles. Collected research articles are reviewed to identify the research gap in the chosen area and then to formulate the research agenda for the same. During the review of the collected research work, articles are verified based on the title, journal of publication, abstract and the detailed research work accomplished. Depending on the relevance of the research work done, a decision was taken to include the research article as part of the literature review. The process of assimilation, synthesize and compilation of the final set of literature review was done and a total of sixty-five research articles were included in the literature review process.

4. Review of Literature on Related Works

The details of the literature review performed is reported under five different categories of review based on the five different keywords.

Review of literature performed on keyword Parallel Sorting Techniques are listed in Table 1.

Table 1 – Scholarly literature review on Parallel Sorting Techniques

Focus area of research	Outcome of the Research	Reference
GPU based Parallel Sorting	The implemented parallel sort achieved five times better speed in comparison with the CUDA library method.	Gupta, S. K., et al., (2023). [1]
Shared Memory Sorting Technique	Block wise approach in parallel sorting provides three times faster to the existing in-place comparison algorithms.	Axtmann, M., et al., (2022). [2]

Segmented Sorting Method	Research results found that fix sorting method requires almost equal amount of time to arrays with equal or different segment sizes.	Schmid, R. F., et al., (2022). [3]
Clustering of Elements	This research paper is based on the performance improvement on NASA lander dataset clustering.	Morrical, N., et al., (2022). [4]
Parallel Divide-and-Conquer Algorithms	Computational complexity of insertion sort is reduced with respect to the sequential procedure.	Ganapathi, P., et al., (2022). [5]
Sorting with Offset-value Coding	Implementation with the help of Google's Napa and F1 Query systems are done to achieve better execution time and scalability.	Do, T., et al., (2023). [6]
Parallel Heat Bath Configuration	An efficient hash function is created using a parallel method to distribute determinants using MPI and OpenMP.	Dang, D. K., et al., (2022). [7]
Parallel Average Partitioning and Sorting Algorithm	Improvement on parallel partitioning and sorting techniques is done to remove the bottleneck problems.	Helal, A. H., et al., (2022). [8]
Meta-Analytical Comparison of Two Sorting Algorithms	The result of comparison done on quick sort and merge sort shows no difference in terms of energy efficiency.	Dlamini, G., et al., (2022). [9]
Parallel Image Encryption Algorithm	Parallel image encryption algorithm is developed with this research work using bitplane scrambling procedure with multiple threads to decrease encryption time.	Song, W., et al., (2023). [10]
Hybrid Parallel Clustering Algorithm	This research work developed a hybrid parallel clustering algorithm that is highly scalable on distributed systems.	Wu, G., et al., (2022). [11]
Sorting Large Data Sets	The research work developed with a model for sorting huge data sets by utilizing subarrays of equal length. Research results show that the mean-based pivot approach is computationally equivalent to median-based approach.	Moghaddam, S. S., et al., (2022). [12]
Parallel Counting Sort Algorithm	Developed an efficient Parallel Counting Sort Algorithm for Optoelectronic Network, that required minimum communication and data exchange.	Gupta, A. (2022). [13]
Method for Reducing Time Complexity	This research work provides a listing of pre-processing approaches for quicksort and insertion sort to improve its execution time. Execution time of the data set with pre-processing approaches is compared with the original sorting method and found the current approach provides better results.	Mubarak, A., et al., (2022). [14]
Parallel Weighted Random Sampling	Developed and implemented parallel and distributed algorithms for building data structures which require sampling single items. Research work shows that tables and sampling with replacement better performance with 158 threads of execution.	Hübschle-Schneider, L., et al., (2022). [15]
Parallel Balanced Binary Trees	This research paper demonstrates that to design a parallel balanced binary search tree single function, join can be used effectively. Algorithm efficiency is verified using C++ Standard Template Library (STL) on numerous input sizes.	Blelloch, G., et al., (2022). [16]
Efficient Parallel Integer Sorting	Developed a high performing CREW PRAM algorithm for integer sorting with less execution time in comparison to the sequential implementation.	Han, Y., et al., (2022). [17]

Review of literature performed on keyword Parallel Searching Algorithms are listed in Table 2.

Table 2 – Scholarly literature review on Parallel Searching Algorithms

Focus area of research	Outcome of the Research	Reference
Local Search Algorithms	Developed two local-search methods for the vertex bisection minimization problem using various graphs.	Tian, X., et al., (2022). [18]
Parallel SVM Algorithm	Implemented a parallel Support Vector Machine (SVM) algorithm also identified advantages and disadvantages of using the same.	Tavara, S. (2019). [19]
Reducing Branch Divergence	The research work aims to reduce branch divergence to enhance the performance of parallel implementation of unit testing.	Bagies, T., et al., (2023). [20]
Sky Sorter Model	Developed a Sky Sorter Model for sorting large scale datasets. It uses the concept of dividing the large datasets into many buckets with sampled splitting.	Zokaee, F., et al., (2022). [21]
Parallel Virtual Screening Software	This research paper performed a review on Parallel Virtual Screening Software's for biopharmaceutical companies. Study shows that parallel implementation of silico virtual screening can escalate the process of drug discovery in affordable time.	Murugan, N. A., et al., (2022). [22]
Dijkstra's Algorithm	This research work proposed an improved version of Dijkstra's algorithm by parallelizing it. Also tested the new method with the existing sequential procedure for the execution comparisons.	Wu, Q., et al., (2015). [23]
Finding All-Pairs Shortest Path	This research paper made the implementation of All-Pairs Shortest Path using parallel Dijkstra Algorithm. It is also used in resolving the problems related to large transportation network problems.	Pradhan, A., et al., (2013). [24]
Dijkstra Algorithm	This research paper suggested a method in identifying the shortest paths in Java Island using the Dijkstra Algorithm.	Amaliah, B., et al., (2016). [25]
Dijkstra's Algorithm	This research paper proposed Dijkstra's Algorithm by parallelizing using CUDA to solve the problems related to dense graphs.	Singh, D., et al., (2016). [26]
Dijkstra's Algorithm	This research paper done with the comparison of Greedy, A* and Dijkstra's Algorithms for finding the Shortest Path problems.	Wayahdi, M. R.,et al., (2021). [27]
Dijkstra's Algorithm	Developed modified Dijkstra's Algorithm for Critical Path Method problem in a project network.	Shankar, N. R., et al., (2010). [28]
Single Source Shortest Path Algorithm	This research paper proposed parallel Single Source Shortest Path Algorithms which are scalable.	Chakaravarthy, V. T., et al., (2016). [29]
Parallel Gauss Jordan Algorithm	This research paper proposed a parallel Gauss Jordan algorithm for matrix inversion using CUDA.	Sharma, G., et al., (2013). [30]
Parallel and Distributed Algorithms	This research work is centered on simple, parallel and distributed algorithms used for Spectral Graph that are Sparse.	Koutis, I., et al., (2016). [31]
Genetic Algorithm	This paper made cooperative parallel grouping genetic algorithms for a specific problem domain.	Kucukyilmaz, T., et al., (2018). [32]
Clustering Algorithm	This research work is based on the development of parallel fuzzy clustering procedures for large graphs using Pregel.	Bhatia, V., et al., (2017). [33]

Parallel Finite Element Algorithm	This research work is based on the study of local and parallel finite element algorithms.	Zheng, H., et al., (2015). [34]
Resultant Elimination Algorithm	This paper focused on the development of resultant elimination algorithms in parallel that are useful in harmonic elimination problems.	Yang, K., et al., (2016). [35]
Parallel Depth First Algorithm	Developed parallel depth first algorithm for number search problem.	Kaneko, T. (2010). [36]
Depth First Algorithm	This research paper developed a parallel Depth First Search Algorithm method, also listing various applications.	Hoki, K., et al., (2013). [37]
Depth First Algorithm	This research paper proposed AND/OR depth-first search algorithmic method for combinatorial optimization.	Otten, L., et al., (2012). [38]
Breadth First Search	This research paper developed a parallel and distributed Breadth First Search procedure on the KEPLER architecture.	Bisson, M., et al., (2015). [39]
Top-Down Breadth-First Search	This paper proposed a parallel path-following phase algorithm based on a breadth-first search approach.	García, L. L., et al., (2020). [40]
Breadth First Search	This research paper developed a breadth first search method on multi-GPU systems, that are efficient and parallel.	Mitsuishi, T., et al., (2016). [41]
Breadth-First Search	This research paper developed an adaptive breadth-first search algorithm that is useful in integrated architectures.	Zhang, F., et al., (2018). [42]

Review of literature performed on keyword Graph Analytics Problem are listed in Table 3.

Table 3 – Scholarly literature review on Graph Analytics Problem

Focus area of research	Outcome of the Research	Reference
Minimum Spanning Tree Algorithm	This research work proposed a fast minimum spanning tree algorithm based on K-means clustering.	Zhong, C., et al., (2015). [43]
Minimum Spanning Tree	This research paper developed an optimization procedure for wind farm cable connection arrangement using optimization minimum spanning tree method.	Hou, P., et al., (2016). [44]
Minimum Spanning Tree	This research work proposed a minimum-loss network configuration using minimum spanning tree problem.	Ahmadi, H., et al., (2015). [45]
Minimum Spanning Tree	This research paper proposed a solution for market basket analysis with minimum spanning trees.	Valle, M. A., et al., (2018). [46]
Ant Colony Optimization	This research paper proposed a solution for the traveling salesman problem using a parallel ant colony optimization method.	Gülcü, Ş., et al., (2018). [47]
Parallel Graph Algorithms	This research paper proved that parallel graph algorithms can be fast and scalable using a theoretical approach.	Dhulipala, L., et al., (2021). [48]
Deep Reinforcement Learning	This research paper proposed a deep reinforcement learning and search method for solving the Rubik's cube problem.	Agostinelli, F., et al., (2019). [49]
Cycle Detection	This research work involves the development of real-time constrained cycle detection for large dynamic graphs.	Qiu, X., et al., (2018). [50]

Clustering Algorithm	This research paper worked on implementation of the Markov clustering algorithm in parallel for large-scale networks.	Azad, A., et al., (2018). [51]
K-Means Clustering Algorithms	This is a literature review paper on the analysis of K-means clustering algorithms.	Ikotun, A. M., et al., (2023). [52]
K-Means Clustering Algorithms	Development of parallel K-Means Clustering Algorithms for cloud assisted IoT.	Mydhili, S. K., et al., (2020). [53]
Parallel Toolkit	Developed a Parallel Toolkit for Large-scale Structure which is open-source and massive parallel.	Hand, N., et al., (2018). [54]
Clustering Algorithm	This research paper proposed a clustering algorithm by using the weighted grid and information entropy based on MapReduce which is parallel.	Yu, X., et al., (2021). [55]

Review of literature performed on keyword Scalable Data Science are listed in Table 4.

Table 4 – Scholarly literature review on Scalable Data Science

Focus area of research	Outcome of the Research	Reference
Climate Change Detection	For the problem of climate change detection, developed a spatial cumulative sum algorithm with big data analytics.	Manogaran, G., et al., (2018). [56]
Big Data Analytics	This paper includes literature review concepts based on integration of big data analytics for value-creation.	Saggi, M. K., et al., (2018). [57]
IoT and Big Data Analytics	This research paper studies smart digital city problems using real-time urban data with the help of IoT and big data analytics tools.	Rathore, M. M., et al., (2018). [58]
Big Data Analytics	This is a survey-based paper based on big data analytics for performing multimedia analytics.	Pouyanfar, S., et al., (2018). [59]
Healthcare Analytics	This is a literature review paper based on big data analytics and its usage in the healthcare system.	Palanisamy, V., et al., (2019). [60]
Machine Learning	This is a literature review paper based on machine learning in Python that focuses on the applications of machine learning in data science.	Raschka, S., et al., (2020). [61]
Social Media Analytics	This is a literature review paper based on the concept of social media big data analytics.	Ghani, N. A., et al., (2019). [62]
Big Data Analytics	This research paper focused on study and implementation of tensor completion algorithms for big data analytics.	Song, Q., et al., (2019). [63]

Review of literature performed on keyword ABCD and SWOT analysis are listed in Table 5.

Table 5 – Scholarly literature review on Analysis

Focus area of research	Outcome of the Research	Reference
ABCD Analysis	This research paper focused on performing ABCD Analysis on high-performance computing.	Kumar, S., et al., (2023). [64]
SWOT Analysis	To perform SWOT analysis based on 5 models for the consortium management.	Srinivasa Rao, Y. et al., (2017). [65]

5. Current Status & New Related Issues

Parallel sorting and searching techniques are commonly used in graph analytics problems to overcome the computational risk while handling large-scale graphs or scalable networks. Parallel algorithms can increase the efficiency of graph analytics problems to optimize the repeated sorting and searching operations involved in these tasks. But as per the current study there is a possibility of research scope for the implementation and testing of various parallel processing APIs OpenMP, MPI and CUDA to enhance the operations involved in graph analytics problems.

6. Ideal Solution, Desired Status & Improvements Required

There is a research need to find execution efficiency of parallel processing APIs OpenMP, MPI and CUDA in graph analytics problems, and how each one of these differs at different types of problems. Also identify the various application domains of scalable data science where graph analytics problems can enhance the operational capabilities by utilizing parallel sorting and searching techniques.

7. Research Gap

Systematic literature review on the topic of Parallel Sorting and Searching Algorithms for Graph Analytics Problems has identified certain research gaps in using parallel sorting and searching techniques in graph analytics problems, which are listed as shown in Table 6.

Table 6 – Identified Research Gap through systematic literature review.

Sr. No.	Area of Literature Review	Identified Research Gap
1.	Parallel Sorting Techniques	**Research Gap 1:** There is a research gap in analyzing the computational performance of different parallel sorting algorithms like Parallel Radix Sort, Parallel Merge Sort, Parallel Quick Sort, Parallel Sample Sort and Parallel Bucket Sort.
2.	Parallel Searching Algorithms	**Research Gap 2:** There is a research gap in analyzing the computational performance of parallel searching algorithms Parallel Dijkstra's Algorithm and Parallel A*. **Research Gap 3:** There is a research gap in analyzing the computational performance of parallel searching algorithms Parallel Depth-First Search and Parallel Breadth-First Search.
3.	Graph Analytics Problem	**Research Gap 4:** There is a research gap in analyzing the computational performance of graph analytics problem Minimum Spanning Tree using different parallel sorting techniques. **Research Gap 5:** There is a research gap in analyzing the computational performance of graph analytics problem Shortest Path Problem. **Research Gap 6:** There is a research gap in analyzing the computational performance graph analytics problems Cycle Detection using Parallel Depth-First Search and Cluster Identification using Parallel Breadth-First Search.

4.	Scalable Data Science	**Research Gap 7:** There is a research gap in using the graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification that use parallel sorting and searching techniques in various domains of scalable data science applications.

8. Research Agendas Based on Research Gap

Associated with various research gaps identified during the literature review, the following research agenda is proposed as shown in Table 7.

Table 7 – Proposed Research Agenda for the Identified Research Gap

Sr. No.	Identified Research Gap	Proposed Research Agenda
1.	**Research Gap 1:** Analyzing the computational performance of different parallel sorting algorithms like Parallel Radix Sort, Parallel Merge Sort, Parallel Quick Sort, Parallel Sample Sort and Parallel Bucket Sort.	(1) To design and implement various parallel sorting algorithms using different parallel processing APIs. (2) To analyze the computational performance of different parallel sorting algorithms.
2.	**Research Gap 2:** Analyzing the computational performance of parallel searching algorithms Parallel Dijkstra's Algorithm and Parallel A*.	(1) To design and implement two parallel searching algorithms Parallel Dijkstra's Algorithm and Parallel A*. (2) To analyze the computational performance of parallel searching algorithms Parallel Dijkstra's Algorithm and Parallel A*.
3.	**Research Gap 3:** Analyzing the computational performance of parallel searching algorithms Parallel Depth-First Search and Parallel Breadth-First Search.	(1) To design and implement two parallel searching algorithms Parallel Depth-First Search and Parallel Breadth-First Search. (2) To analyze the computational performance of parallel searching algorithms Parallel Depth-First Search and Parallel Breadth-First Search.
4.	**Research Gap 4:** Analyzing the computational performance of graph analytics problem Minimum Spanning Tree using different parallel sorting techniques.	(1) To design and implement graph analytics problem Minimum Spanning Tree. (2) To analyze the computational performance of graph analytics problem Minimum Spanning.
5.	**Research Gap 5:** Analyzing the computational performance of graph analytics problem Shortest Path Problem.	(1) To design and implement graph analytics problem Shortest Path Problem. (2) To analyze the computational performance of graph analytics problem Shortest Path Problem.
6.	**Research Gap 6:** Analyzing the computational performance graph analytics problems Cycle Detection using Parallel Depth-First Search and Cluster Identification using Parallel Breadth-First Search.	(1) To design and implement graph analytics problem of Cycle Detection and Cluster Identification. (2) To analyze the computational performance of graph analytics problem of Cycle Detection and Cluster Identification.
7.	**Research Gap 7:** Using the graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification that use parallel sorting and searching techniques in various domains of scalable data science applications.	(1) To identify sources of scalable data sets from the existing graph data sets. (2) To apply graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification on identified scalable graph data sets.

	(3) To analyze the outcome of different graph analytics problems on scalable data sets.

9. SWOT Analysis of Research Agendas

SWOT analysis is used in finding Strengths-Weaknesses-Opportunities-Threats of a particular business and associated policies [65]. SWOT analysis is performed on the proposed research agendas to identify the Strengths-Weaknesses-Opportunities-Threats with respect to the research gap identified. This is useful for finalizing the research agendas that can be later mapped to different research objectives of the proposed research work.

9.1. Strengths of the Research Agendas

(1) The research agenda is created as a sequential step of accomplishments, where each one can contribute to the completion of the successor.
(2) The agenda clearly focuses on parallel sorting and searching techniques and its usage in graph analytics problems.
(3) The agenda is well defined to form the research objectives for the research proposal.
(4) It is proposed by considering the time bound completion of the research proposal, which is technically and economically feasible.

9.2. Weaknesses of the Research Agendas

(1) The accuracy of the study depends on the model or algorithm designed in parallel.
(2) There may be alternate research findings possible in due course of time for the parallel implementation.

9.3. Opportunities of the Research Agendas

(1) It will help in implementing various sorting and searching techniques using different parallel processing APIs.
(2) Different graph analytics problems can be solved using the designed parallel sorting and searching techniques and its execution efficiency can be studied and tested.
(3) The use of graph analytics problems in scalable data science can be implemented and verified, which will help various application domain stakeholders to receive a better solution.

9.4. Threats of the Research Agendas

(1) Drastic change in the field of parallel processing APIs can disrupt the current research study.
(2) There can be alternate methods to enhance the performance of graph analytics problems in scalable data science.

10. Final Research Proposal/Problem in Chosen Topic

With the insight of systematic literature review and considering the research gap and research agenda in this paper, following is our research proposal:

10.1. Title

Design and Implementation of Parallel Sorting and Searching Algorithms for Graph Analytics Problems in Scalable Data Science

10.2. Purpose

There is a need to design various parallel sorting and searching algorithms that can be used in graph analytics problems. Analysis of different parallel sorting algorithms will help in using the efficient one as per the chosen problem domain to optimize the solution. Different searching algorithms need to be designed using parallel APIs that can be used in specific graph analytics problems like Shortest Path Problem, Cycle Detection, and Cluster Identification. These graph analytics methods along with parallel techniques can be used in solving problems related to scalable data science.

10.3. Research Objectives

(1) To design and implement various parallel sorting algorithms like Parallel Radix Sort, Parallel Merge Sort, Parallel Quick Sort, Parallel Sample Sort and Parallel Bucket Sort using parallel processing APIs OpenMP, MPI and CUDA.
(2) To analyze the computational performance of different parallel sorting algorithms that are designed and implemented.
(3) To design and implement graph analytics problem Minimum Spanning Tree using different parallel sorting techniques.
(4) To analyze the computational performance of graph analytics problem Minimum Spanning Tree using different parallel sorting techniques.
(5) To design and implement two parallel searching algorithms Parallel Dijkstra's Algorithm and Parallel A* using parallel processing APIs OpenMP, MPI and CUDA.
(6) To analyze the computational performance of parallel searching algorithms Parallel Dijkstra's Algorithm and Parallel A* that are designed and implemented.
(7) To design and implement graph analytics problem Shortest Path Problem using Parallel Dijkstra's Algorithm and Parallel A* searching techniques.
(8) To analyze the computational performance of graph analytics problem Shortest Path Problem using both Parallel Dijkstra's Algorithm and Parallel A* searching techniques with different graphs.
(9) To design and implement two parallel searching algorithms Parallel Depth-First Search and Parallel Breadth-First Search using parallel processing APIs OpenMP, MPI and CUDA.
(10) To analyze the computational performance of parallel searching algorithms Parallel Depth-First Search and Parallel Breadth-First Search that are designed and implemented.
(11) To design and implement graph analytics problems of Cycle Detection using Parallel Depth-First Search searching techniques.
(12) To design and implement graph analytics problems of Cluster Identification using Parallel Breadth-First Search searching techniques.
(13) To analyze the computational performance of graph analytics problems Cycle Detection using Parallel Depth-First Search and Cluster Identification using Parallel Breadth-First Search using different graphs.
(14) To apply graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification in various applications of data science to achieve scalable data science.

10.4. Proposed Methodology

Our research proposal follows two Computer Science Research Methodologies:

(1) Theoretical Computer Science Research Methodology
and
(2) Experimental Computer Science Research Methodology

(1) **Theoretical Computer Science Research Methodology**

Theoretical Computer Science research methodology follows classical methodology in construction principles as logical structures with objects (axioms) and operations (rules) for deriving theorems. Fundamental ideas in this computing research method are to design conceptual or formal models with data models, algorithmic procedures and analysis of the implemented algorithm efficiency.

Major operational themes in Theoretical Computer Science include Iteration, Induction and Recursion.

- Iteration - Iteration is the method of repeating a sequence of instructions to perform the repeated operations. This includes constructs such as for or while statements.
- Induction - Inductive classifications and proofs use basic and inductive steps to comprehend all likely cases.
- Recursion - The definition of recursive procedures contains calls to the same procedure either directly or indirectly. It can be considered as self-definition, in which a model is defined in terms of itself.

Theoretical Computer Science Research Methodology will be followed in achieving the research objectives of designing and implementing various parallel sorting and searching procedures and in designing and implementing graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification.

(2) **Experimental Computer Science Research Methodology**

Experimental Computer Science Research Methodology is used to apprehend the method of information processes, to observe phenomena, to articulate explanations and theories, and to test them. Experiments are used in both theory testing and for exploration. These are used in fields of study where theory and deductive analysis are not directly possible.

Experimental Computer Science Research Methodology will be followed in achieving the research objectives of analyzing the computational performance of various parallel sorting and searching procedures and in analyzing graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification using different graphs. These techniques will be used to experiment in various applications of data science to achieve scalable data science.

10.5. Results and Findings from Study

For the better understanding and representation of the research outcomes, analysis of various objectives will be represented in tabular, diagrammatic, and figurative form.

10.6. Conclusion & Implications

The conclusion will highlight the various analysis of sorting and searching techniques that are designed and implemented in parallel. Also, it illustrates the use of these parallel sorting and searching techniques in solving graph analytics problems like Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification. The implications of the outcome to the stakeholders will be discussed with the use of parallel sorting and searching techniques in graph analytics problems to solve scalable data science problems. The scope of further studies will also be listed.

10.7. Research Scope and Constraints

The research proposal is limited to a few sorting and searching techniques that can be designed and implemented in parallel and used in solving graph analytics problems Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification. There are more sorting and searching techniques that can be designed and implemented in parallel and can be used in solving various problems in graph analytics. The scope of the research proposal is limited to the research objectives listed above in 10.3 due to the time constraints. Further, more parallel algorithms can be designed and implemented and their use in graph analytics problems can be explored and its usage in scalable data science.

11. ABCD Analysis of Chosen Research Proposal

ABCD analysis done on a particular industry can provide an extensive list of the industry analysis results with various advantages, benefits, constraints, and disadvantages of the specific industry as a system matrix [64]. ABCD analysis is performed on the proposed research topic titled ***"Design and Implementation of Parallel Sorting and Searching Algorithms for Graph Analytics Problems in Scalable Data Science"*** to identify the advantages, benefits, constraints, and disadvantages of the given research proposal. This helps the researcher to have an overall understanding of the research proposal and its feasibility of implementing the proposal.

11.1. Advantages of the research proposal

(1) Design and implementation of various parallel sorting algorithms like Parallel Radix Sort, Parallel Merge Sort, Parallel Quick Sort, Parallel Sample Sort and Parallel Bucket Sort will be done that can be used in solving various computational problems that needs sorting techniques.

(2) Design and implementation of parallel searching algorithms like Parallel Dijkstra's Algorithm, Parallel A*, Parallel Depth-First Search and Parallel Breadth-First Search will be done that can be used in solving various computational problems that need searching techniques.

11.2. Benefits of the research proposal

(1) Computational performance analysis of different parallel sorting algorithms provides the users to select the optimum sorting techniques to the required problems to be solved.
(2) Implemented graph analytics problem Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification using parallel sorting and searching techniques will provide optimum results in comparison with the sequential algorithms. Thus, it provides users with applications of scalable data science.
(3) Listing various uses of graph analytics problem Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification will help researchers and data scientists to use these techniques in solving the upcoming problems that require similar techniques.
(4) The use of graph analytics problems in scalable data science results are provided with real-time graph data sets.

11.3. Constraints of the research proposal

(1) The proposal is time-bound due to which few sorting and searching techniques are designed and implemented in parallel.
(2) There are more graph analytics problems that can be solved using parallel sorting and searching techniques than listed in the research proposal.
(3) The proposal will address the applications of graph analytics problem Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification in scalable data science, but it will not be limited to the listing.

11.4. Disadvantages of the Research Proposal

(1) More graph analytics problems can be optimized using parallel sorting and searching techniques that are not mentioned in this research proposal.
(2) There can be alternative computational ways to optimize the graph analytics problems than parallel implementation.
(3) The execution efficiency of parallel algorithms also depends on the multiple processors that are involved in solving the problem and its effective communication.

12. Suggestions to Implement Research Activities According to the Proposal

To implement the research activities according to the research proposal it will follow two computational science research methodologies: Theoretical Computer Science Research Methodology and Experimental Computer Science Research Methodology.

(1) Steps in Theoretical Computer Science Research Methodology

- Design and implement various parallel sorting algorithms Parallel Radix Sort, Parallel Merge Sort, Parallel Quick Sort, Parallel Sample Sort and Parallel Bucket Sort.
- Design and implement four parallel searching algorithms Parallel Dijkstra's Algorithm, Parallel A*, Parallel Depth-First Search and Parallel Breadth-First Search.
- Design and implement graph analytics problem Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification.

(2) Steps in Experimental Computer Science Research Methodology

- Computational performance analysis of different parallel sorting algorithms that are designed and implemented.
- Computational performance analysis of different parallel searching algorithms that are designed and implemented.
- Computational performance analysis of graph analytics problem Minimum Spanning Tree, Shortest Path Problem, Cycle Detection and Cluster Identification that are designed and implemented.
- Apply graph analytics problems Minimum Spanning Tree Shortest Path Problem, Cycle Detection and Cluster Identification in various applications of data science to achieve scalable data science.

(3) Data Collection

Primary data will be collected by executing the implemented algorithms using random graphs and random numbers. Multiple executions of the implemented algorithms will be performed by taking different input graphs and different sets of numbers. The same graphs or same set of numbers will be used while executing different algorithms to find the execution time of the algorithm. All the graph analytics problems mentioned in the research objectives will be executed in parallel and sequential methods and collect the data as execution time required for both the methods. Scalable data sets from the existing graph data sets will be used for various experimental processes of graph analytics problems.

(4) Data Analysis

Data analysis will be performed on the collected data and produce the result as the optimum algorithmic method for parallel sorting and searching techniques. Data analysis will be performed on graph analytics problems to understand the execution efficiency of using parallel sorting and searching techniques for graph analytics problems.

(5) Result, Interpretation and Conclusion

For the better understanding and representation of the research outcomes, analysis of various objectives will be represented in tabular, diagrammatic, and figurative form.

(6) Bibliography and References

Related literature books, websites, research journals and papers will be acknowledged that will help someone who wants to do further research in the associated area of parallel algorithms and its usage in graph analytics problems.

13. Limitations of the Proposal

There are a set of limitations that are identified for the proposed research work:

(1) There can be more parallel sorting techniques that can be implemented using parallel processing APIs than that are mentioned in the proposed research work.

(2) There can be more parallel searching techniques that can be implemented using parallel processing APIs than that are mentioned in the proposed research work.

(3) Analysis of parallel sorting and searching techniques can be done through theoretical methods other than that is proposed in the current study. In the proposed study of research analytical methods of experimentation using different sets of data sets and random numbers for the sorting and searching methods are proposed.

(4) There are numerous graph analytics problems that can be optimized using parallel sorting and searching methods. There can be other parallel execution procedures as well for the optimization of graph analytics problems.

(5) To make data science handle scalable data there are alternative methods as well, than that are proposed in this research work.

14. Conclusion

The purpose of this research work is to perform a literature review on the major four domains of studies like Parallel Sorting Techniques, Parallel Searching Algorithms, Graph Analytics Problem and Scalable Data Science. The aim of this research is to identify various research gaps in the chosen domain of study and map it to the corresponding research agendas. SWOT analysis is performed on the research agendas that helped in proposing a new study in the domain of study. As a result, the outcome of this literature review paper has resulted in a research proposal of Design and Implementation of Parallel Sorting and Searching Algorithms for Graph Analytics Problems in Scalable Data Science. Also, we have identified fourteen research objectives for the research proposal. The methodologies to be followed during the process of proposed research work are identified as Theoretical Computer Science Research Methodology and Experimental Computer Science Research Methodology. Listing of ABCD analysis is performed on the research proposal that provides an insight into the topic of study, its feasibility of implementation, time-bound completion possibility and social, economic, knowledge-based relevance of the study. Various suggestions to implement our research proposal are specified like research methodologies to be followed, method of data collection, sources of data collection, procedure to be followed during the data analysis, representation of results, interpretation, and conclusion.

References

[1] Gupta, S. K., Singh, D. P., & Choudhary, J. (2023). New GPU Sorting Algorithm Using Sorted Matrix. Procedia Computer Science, 218(1), 1682-1691.

[2] Axtmann, M., Witt, S., Ferizovic, D., & Sanders, P. (2022). Engineering in-place (shared-memory) sorting algorithms. ACM Transactions on Parallel Computing, 9(1), 1-62.

[3] Schmid, R. F., Pisani, F., Cáceres, E. N., & Borin, E. (2022). An evaluation of fast segmented sorting implementations on GPUs. Parallel Computing, 110(1), 102889-102910.

[4] Morrical, N., Sahistan, A., Güdükbay, U., Wald, I., & Pascucci, V. (2022). Quick clusters: A GPU-parallel partitioning for efficient path tracing of unstructured volumetric grids. IEEE Transactions on Visualization and Computer Graphics, 29(1), 537-547.

[5] Ganapathi, P., & Chowdhury, R. (2022). Parallel Divide-and-Conquer Algorithms for Bubble Sort, Selection Sort, and Insertion Sort. The Computer Journal, 65(10), 2709-2719.

[6] Do, T., & Graefe, G. (2023). Robust and efficient sorting with offset-value coding. ACM Transactions on Database Systems, 48(1), 1-23.

[7] Dang, D. K., Kammeraad, J. A., & Zimmerman, P. M. (2022). Advances in parallel heat bath configuration interaction. The Journal of Physical Chemistry A, 127(1), 400-411.

[8] Helal, A. H., & Shaheen, M. E. (2022). Improvement of parallel average partitioning and sorting algorithm with decreasing single point of failure and bottleneck. J. Theor. Appl. Inf. Technol, 100(1), 25-46.

[9] Dlamini, G., Jolha, F., Kholmatova, Z., & Succi, G. (2022). Meta-analytical comparison of energy consumed by two sorting algorithms. Information Sciences, 582(1), 767-777.

[10]Song, W., Fu, C., Zheng, Y., Tie, M., Liu, J., & Chen, J. (2023). A parallel image encryption algorithm using intra bitplane scrambling. Mathematics and Computers in Simulation, 204(1), 71-88.

[11]Wu, G., Cao, L., Tian, H., & Wang, W. (2022). HY-DBSCAN: A hybrid parallel DBSCAN clustering algorithm scalable on distributed-memory computers. Journal of Parallel and Distributed Computing, 168(1), 57-69.

[12]Moghaddam, S. S., & Moghaddam, K. S. (2022). A general framework for sorting large data sets using independent subarrays of approximately equal length. IEEE Access, 10(1), 11584-11607.

[13]Gupta, A. (2022). Optimized Parallel Counting Sort Algorithm for Distinct Numeric Values on Biswapped Hyper Hexa-Cell Optoelectronic Network. International Journal of Computer Network and Information Security, 13(1), 69-83.

[14]Mubarak, A., Iqbal, S., Rasool, Q., Asghar, N., Faujdar, N., & Rauf, A. (2022). Preprocessing: A method for reducing time complexity. Journal of Computing & Biomedical Informatics, 4(01), 104-117.

[15]Hübschle-Schneider, L., & Sanders, P. (2022). Parallel weighted random sampling. ACM Transactions on Mathematical Software (TOMS), 48(3), 1-40.

[16]Blelloch, G., Ferizovic, D., & Sun, Y. (2022). Joinable parallel balanced binary trees. ACM Transactions on Parallel Computing, 9(2), 1-41.

[17]Han, Y., & He, X. (2022). More Efficient Parallel Integer Sorting. International Journal of Foundations of Computer Science, 33(05), 411-427.

[18]Tian, X., Ouyang, D., Sun, R., Zhou, H., & Zhang, L. (2022). Two efficient local search algorithms for the vertex bisection minimization problem. Information Sciences, 609(1), 153-171.

[19]Tavara, S. (2019). Parallel computing of support vector machines: a survey. ACM Computing Surveys (CSUR), 51(6), 1-38.

[20]Bagies, T., Le, W., Sheaffer, J., & Jannesari, A. (2023). Reducing branch divergence to speed up parallel execution of unit testing on GPUs. The Journal of Supercomputing, 1(1)1-35.

[21]Zokaee, F., Chen, F., Sun, G., & Jiang, L. (2022). Sky-Sorter: A Processing-in-Memory Architecture for Large-Scale Sorting. IEEE Transactions on Computers, 72(2), 480-493.

[22] Murugan, N. A., Podobas, A., Gadioli, D., Vitali, E., Palermo, G., & Markidis, S. (2022). A review on parallel virtual screening softwares for high-performance computers. Pharmaceuticals, 15(1), 63.

[23] Wu, Q., Qin, G., & Li, H. (2015). An Improved Dijkstra's algorithm application to multi-core processors. Metallurgical & Mining Industry, 9(1), 76-81.

[24]Pradhan, A., & Mahinthakumar, G. (2013). Finding all-pairs shortest path for a large-scale transportation network using parallel Floyd-Warshall and parallel Dijkstra algorithms. Journal of computing in civil engineering, 27(3), 263-273.

[25]Amaliah, B., Fatichah, C., & Riptianingdyah, O. (2016). Finding the shortest paths among cities in Java Island using node combination based on Dijkstra algorithm. International Journal on Smart Sensing and Intelligent Systems, 9(4), 2219-2236.

[26]Singh, D., & Khar, N. (2016). Modified Dijkstra's algorithm for dense graphs on GPU using CUDA. Indian Journal of Science and Technology, 9(33), 1-9.

[27]Wayahdi, M. R., Ginting, S. H. N., & Syahputra, D. (2021). Greedy, A-Star, and Dijkstra's algorithms in finding shortest path. International Journal of Advances in Data and Information Systems, 2(1), 45-52.

[28]Shankar, N. R., & Sireesha, V. (2010). Using modified Dijkstra's algorithm for critical path method in a project network. International Journal of Computational and Applied Mathematics, 5(2), 217-225.

[29]Chakaravarthy, V. T., Checconi, F., Murali, P., Petrini, F., & Sabharwal, Y. (2016). Scalable single source shortest path algorithms for massively parallel systems. IEEE Transactions on Parallel and Distributed Systems, 28(7), 2031-2045.

[30]Sharma, G., Agarwala, A., & Bhattacharya, B. (2013). A fast parallel Gauss Jordan algorithm for matrix inversion using CUDA. Computers & Structures, 128(1), 31-37.

[31]Koutis, I., & Xu, S. C. (2016). Simple parallel and distributed algorithms for spectral graph sparsification. ACM Transactions on Parallel Computing (TOPC), 3(2), 1-14.

[32]Kucukyilmaz, T., & Kiziloz, H. E. (2018). Cooperative parallel grouping genetic algorithm for the one-dimensional bin packing problem. Computers & Industrial Engineering, 125(1), 157-170.

[33]Bhatia, V., & Rani, R. (2017). A parallel fuzzy clustering algorithm for large graphs using Pregel. Expert Systems with Applications, 78(1), 135-144.

[34]Zheng, H., Yu, J., & Shi, F. (2015). Local and parallel finite element algorithm based on the partition of unity for incompressible flows. Journal of Scientific Computing, 65(2), 512-532.

[35]Yang, K., Chen, L., Zhang, J., Hao, J., & Yu, W. (2016). Parallel resultant elimination algorithm to solve the selective harmonic elimination problem. IET Power Electronics, 9(1), 71-80.

[36]Kaneko, T. (2010, July). Parallel depth first proof number search. In Proceedings of the AAAI Conference on Artificial Intelligence, 24(1), 95-100.

[37]Hoki, K., Kaneko, T., Kishimoto, A., & Ito, T. (2013). Parallel dovetailing and its application to depth-first proof-number search. ICGA Journal, 36(1), 22-36.

[38]Otten, L., & Dechter, R. (2012). Anytime AND/OR depth-first search for combinatorial optimization. Ai Communications, 25(3), 211-227.

[39]Bisson, M., Bernaschi, M., & Mastrostefano, E. (2015). Parallel distributed breadth first search on the Kepler architecture. IEEE Transactions on Parallel and Distributed Systems, 27(7), 2091-2102.

[40]García, L. L., Arellano, A. G., & Cruz-Santos, W. (2020). A parallel path-following phase unwrapping algorithm based on a top-down breadth-first search approach. Optics and Lasers in Engineering, 124(1), 105827-105841.

[41]Mitsuishi, T., Suzuki, J., Hayashi, Y., Kan, M., & Amano, H. (2016). Breadth first search on cost-efficient multi-GPU systems. ACM SIGARCH Computer Architecture News, 43(4), 58-63.

[42]Zhang, F., Lin, H., Zhai, J., Cheng, J., Xiang, D., Li, J., ... & Du, X. (2018). An adaptive breadth-first search algorithm on integrated architectures. The Journal of Supercomputing, 74(1), 6135-6155.

[43]Zhong, C., Malinen, M., Miao, D., & Fränti, P. (2015). A fast minimum spanning tree algorithm based on K-means. Information Sciences, 295(1), 1-17.

[44]Hou, P., Hu, W., & Chen, Z. (2016). Optimisation for offshore wind farm cable connection layout using adaptive particle swarm optimisation minimum spanning tree method. IET Renewable Power Generation, 10(5), 694-702.

[45]Ahmadi, H., & Martí, J. R. (2015). Minimum-loss network reconfiguration: A minimum spanning tree problem. Sustainable Energy, Grids and Networks, 1(1), 1-9.

[46]Valle, M. A., Ruz, G. A., & Morrás, R. (2018). Market basket analysis: Complementing association rules with minimum spanning trees. Expert Systems with Applications, 97(1), 146-162.

[47]Gülcü, Ş., Mahi, M., Baykan, Ö. K., & Kodaz, H. (2018). A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. Soft Computing, 22(1), 1669-1685.

[48]Dhulipala, L., Blelloch, G. E., & Shun, J. (2021). Theoretically efficient parallel graph algorithms can be fast and scalable. ACM Transactions on Parallel Computing (TOPC), 8(1), 1-70.

[49]Agostinelli, F., McAleer, S., Shmakov, A., & Baldi, P. (2019). Solving the Rubik's cube with deep reinforcement learning and search. Nature Machine Intelligence, 1(8), 356-363.

[50]Qiu, X., Cen, W., Qian, Z., Peng, Y., Zhang, Y., Lin, X., & Zhou, J. (2018). Real-time constrained cycle detection in large dynamic graphs. Proceedings of the VLDB Endowment, 11(12), 1876-1888.

[51]Azad, A., Pavlopoulos, G. A., Ouzounis, C. A., Kyrpides, N. C., & Buluç, A. (2018). HipMCL: a high-performance parallel implementation of the Markov clustering algorithm for large-scale networks. Nucleic acids research, 46(6), 33-53.

[52]Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622(1), 178-210.

[53]Mydhili, S. K., Periyanayagi, S., Baskar, S., Shakeel, P. M., & Hariharan, P. R. (2020). Retracted article: machine learning based multi scale parallel K-means++ clustering for cloud assisted internet of things. Peer-to-peer Networking and Applications, 13(6), 2023-2035.

[54]Hand, N., Feng, Y., Beutler, F., Li, Y., Modi, C., Seljak, U., & Slepian, Z. (2018). nbodykit: An open-source, massively parallel toolkit for large-scale structure. The Astronomical Journal, 156(4), 160-181.

[55]Yu, X., Zeng, F., Mwakapesa, D. S., Nanehkaran, Y. A., Mao, Y. M., Xu, K. B., & Chen, Z. G. (2021). DBWGIE-MR: A density-based clustering algorithm by using the weighted grid and information entropy based on MapReduce. Journal of Intelligent & Fuzzy Systems, 40(6), 10781-10796.

[56]Manogaran, G., & Lopez, D. (2018). Spatial cumulative sum algorithm with big data analytics for climate change detection. Computers & Electrical Engineering, 65(1), 207-221.

[57]Saggi, M. K., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. Information Processing & Management, 54(5), 758-790.

[58]Rathore, M. M., Paul, A., Hong, W. H., Seo, H., Awan, I., & Saeed, S. (2018). Exploiting IoT and big data analytics: Defining smart digital city using real-time urban data. Sustainable cities and society, 40(1), 600-610.

[59]Pouyanfar, S., Yang, Y., Chen, S. C., Shyu, M. L., & Iyengar, S. S. (2018). Multimedia big data analytics: A survey. ACM computing surveys (CSUR), 51(1), 1-34.

[60]Palanisamy, V., & Thirunavukarasu, R. (2019). Implications of big data analytics in developing healthcare frameworks–A review. Journal of King Saud University-Computer and Information Sciences, 31(4), 415-425.

[61]Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), 193-221.

[62]Ghani, N. A., Hamid, S., Hashem, I. A. T., & Ahmed, E. (2019). Social media big data analytics: A survey. Computers in Human behavior, 101(1), 417-428.

[63]Song, Q., Ge, H., Caverlee, J., & Hu, X. (2019). Tensor completion algorithms in big data analytics. ACM Transactions on Knowledge Discovery from Data (TKDD), 13(1), 1-48.

[64]Kumar, S., & Kunte, R. S. R. (2023). ABCD Analysis of Industries Using High-Performance Computing. International Journal of Case Studies in Business, IT and Education (IJCSBE), 7(2), 448-465.

[65]Srinivasa Rao, Y. (2017). C5 model for the consortium management: SWOT analysis. Library management, 38(4), 248-262.

******