IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 12, Issue 08, October 2025 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1208.07

# Parallel PageRank Computation with AI-Driven Fairness Analysis: A Web-Based Framework

**Sivas S**
Department of Computer Science and Engineering
The Oxford College of Engineering, Bengaluru, India
ss6456462@gmail.com
**ShivaShankar S**
Department of Computer Science and Engineering,
The Oxford College of Engineering, Bengaluru, India
shankar8867136538@gmail.com

**Publication History**
Manuscript Reference: IRJCS/RS/Vol.12/Issue08/OCCS10086
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.12/Issue08/OCCS10086
Received:23,October 2025, Revised: 09, October 2025, Accepted: 31October 2025 Published Online: 21November 2025
https://www.irjcs.com/volumes/Vol12/iss-08/07.OCCS10086.pdf
**Article Citation:Sivas,ShivaShankar** (2025), Parallel PageRank Computation with AI-Driven Fairness Analysis: A Web-Based Framework**,**IRJCS: International Research Journal of Computer Science, Volume 12, Issue 08 of 2025 pages 326-340 **doi:>** https://doi.org/10.26562/irjcs.2025.v1208.07
**BibTeX Sivas@2025Parallel**
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2025, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1208.07 The journal's official abbreviation is IRJCS.
Orcid: https://orcid.org/0009-0004-9398-7488


**Abstract**: PageRank remains a cornerstone algorithm for ranking nodes in large-scale networks, yet modern challenges demand improvements in both scalability and fairness. Traditional implementations are computationally expensive for billions of nodes and often exhibit biases, favoring highly connected entities while suppressing emerging or underrepresented nodes. This paper introduces a novel framework that integrates parallel PageRank computation with an AI-driven fairness analysis module, deployed as a professional web application. The frame- work employs MPI, OpenMP, and CUDA for distributed and accelerated execution, while machine learning models evaluate rankingbiasthroughfairnessmetricssuchasexposureparityand diversity. Experimental results demonstrate significant reductions in execution time and the ability to highlight imbalances in rankings. The system further provides actionable recommendationsto mitigate bias, ensuring both efficiency and ethical outcomes. This unique combination of high-performance computing andAI fairness analysis offers a scalable and socially responsible approach to graph-based ranking systems.
**IndexTerms**: PageRank, Parallel Computing, MPI, OpenMP, CUDA, Fairness in AI, Machine Learning, Web Applications

## I. INTRODUCTION

In the era of big data and complex networks, ranking algorithms have become indispensable for organizing and prioritizing information across diverse domains such as search engines, social media, academic databases, and e-commerce platforms. Among these algorithms, PageRank, introduced by Brin and Page in 1998, has remained a seminal method for evaluating the relative importance of nodes in a directed graph. PageRank assigns a probabilistic score to each node based on the structure of incoming and outgoing links, providing a measure of authority or influence within the network. This metric has proven highly effective in numerous applications, including web search, citation analysis, recommendation systems, and social network evaluation. Despite its success, two major challenges persist in the ap- plication of PageRank: computational scalability and fairness. Computational scalability: Traditional PageRank computations rely on iterative matrix operations that can become prohibitively expensive as network sizes grow to billions of nodes and edges. For example, Visali *et al.*[1] conduct eda comprehensive performance analysis of a parallelized PageRank algorithm using Open MP, MPI, and CUDA, demonstrating that heterogeneous CPU GPU computation can significantly reduce execution time by leveraging hybrid parallelism. Similarly, Sahu [3] introduced a lock-free computation approach for dynamic graphs, showing substantial reductions in synchronization over head. In another work, Sahu[7] presented an efficient GPU implementation of the dynamic DF-P PageRank, achieving notable speedups on incrementally expanding graphs. Shovon *et al.* [5] extended this idea to multinode, multi-GPU frame works using Datalog, providing further scalability in distributed systems. Likewise, Fa¨rber *et al.* [6] explored the integration of graph machine learning pipelines through AutoRDF2GML to improve graph computation efficiency. Collectively, these works highlight that parallelism through OpenMP, MPI, and CUDA can drastically enhance the scalability of PageRank across large-scale datasets. Fairness in ranking outcomes: Beyond scalability, fairness in ranking outcomes has emerged as a critical concern. Traditional PageRank tends to favor nodes with high connectivity, inadvertently suppressing smaller, emerging, or under- represented nodes.

This bias can have significant ethical and societal implications. For example, in search engine rankings, minority-owned businesses or new entrants may receive lower visibility despite relevance. In academic citation networks, early-career researchers may be overlooked in favor of established authors with higher citation counts. Similarly, in social media content ranking, influential users dominate visibility, potentially marginalizing diverse voices.

Several recent studies have investigated fairness in ranking and search systems. Rastogi and Joachims [2] introduced a fairness framework under disparate uncertainty, proposing methods to ensure equitable exposure across different demographic groups even when uncertainty in relevance estimation exists. Fang *et al.* [4] presented a comprehensive survey on fairness in search systems, categorizing fairness concerns and mitigation strategies across query, document, and ranking levels. Balagopalan *et al.* [9] proposed polarity-aware distribution-based fair ranking methods to maintain equitable exposureamongdiversecontentcategories,whileBalzotti *et al.* [10] developed R-Fairness, a framework for assessing fairness of rankings in subjective datasets. Together, these works emphasize that fairness must be explicitly addressed to ensure ethical ranking outcomes in both static and dynamic contexts. The convergence of parallel computing and fairness-aware

Analysis presents a unique opportunity. A high-performance PageRank computation framework can address scalability, while embedding fairness evaluation ensures ethical and equitable outcomes. Although several studies have explored fairness in machine learning and recommendation systems, integration with graph-based ranking algorithms remains

limited. The challenge lies notonly in detecting biases but alsoin providing actionable interventions to mitigate them without compromising computational efficiency or ranking relevance. Motivated by these challenges, this paper proposes a hybrid frameworkthatunitesparallelPageRankcomputationwith AI-drivenfairnessanalysis.Theproposedsystemleverages multi-level parallelism, including MPI for cluster-based computation, OpenMP for multi-core CPU execution, and CUDA for GPU acceleration. Once the PageRank scores are computed, a machine learning module evaluates fairness metrics across predefined node groups and generates corrective recommendations. The entire workflow is encapsulated within a professional web application, enabling users top load graphs, monitor computations, visualize rankings, and inspect fairness outcomes in real time.

The objectives of this research are three fold:

1) To design and implement a scalable, parallel PageRank framework capable of handling large-scale graph datasets efficiently.
2) To integrate AI-driven fairness analysis that quantitatively assesses biases and provides actionable suggestions for ranking adjustments.
3) To deploy the framework as a user friendly web application that allows seamless interaction, visualization, and decision support for diverse users.

By combining these components, the proposed framework addresses both computational and ethical dimensions of modern ranking systems. The system enables organizations and researchers to analyze large networks efficiently while ensuring fair and transparent outcomes. In the following sections, we provide an overview of related work, describe the system architecture, detail the parallel PageRank computation and fairness evaluation, and present experimental results demonstrating the efficacy of the proposed approach.

### A. Motivation and Problem Statement

Traditional PageRank is purely structural and ignores con- textual fairness. This framework bridges the gap by:

- Reducing computational time through multilevel parallelism [1],[3],[7],[5].
- Incorporating fairness metrics to detect bias [2],[4],[9],[10].
- Providing actionable recommendations for equitable visibility.

### B. Contributions

The proposed research makes several significant contributions toward enhancing the scalability and fairness of ranking algorithms. First, this work integrates a hybrid parallel computing architecture for PageRank computation, leveraging MPI, Open MP, and CUDA to exploit distributed, shared- memory, and GPU-based parallelism respectively. This integration, inspired by recent advances in parallel PageRank and performanceoptimizationframeworks[1],[3],[7],[5],enables efficient processing of large-scale graph datasets by distributing workloads across multiple processing units, thereby minimizing computation time and achieving faster convergence. The proposed multi-level approach ensures high scalability and adaptability across heterogeneous computing environments. Second, the system incorporates an AI-driven fairness analysis module that quantitatively evaluates and mitigates biases present in traditional PageRank outputs. This module employs fairness metrics such as *exposure parity*, *diversity index*, and *normalized rank difference*, drawing on state-of- the-art fairness frameworks in ranking and recommendation systems [2],[4],[9],[10]. These measures collectively assess whether the visibility and influence scores are equitably dis- tributed among diverse node groups and offer corrective Suggestions for improving fairness without compromising ranking relevance. Third, the entire framework is encapsulated within a professional, web-based application that enables real-time interaction, visualization, and decision making. This application allows users to upload graph datasets, monitor parallel computations, visualize ranking outcomes, and analyze fairness metrics interactively through an intuitive dashboard. The web interface ensures accessibility and usability for researchers, developers, and organizations. Furthermore, the system undergoes extensive experimental evaluation on both synthetic and real-world datasets to validate its performance and fairness outcomes. The results demonstrate substantial improvements in computational efficiency, fairness awareness, and interpretability compared to traditional PageRank implementations. Finally, this research delivers a unified and Web accessible platform that integrates visualization, reporting, and inter- active analysis.

IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 12, Issue 08, October 2025 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1208.07

The system not only bridges the gap between performance and ethics in ranking computations but also provides a practical, deployable tool that supports fair, transparent, and efficient graph-based analytics for a wide range of applications.

## II. RELATEDWORK

The problem of ranking nodes in large-scale networks has been extensively studied over the past two decades, primarily focusing on improving computational efficiency and, more recently, incorporating fairness considerations. This section provides a comprehensive review of existing approaches, categorized into three main areas: (i) parallel and distributed PageRank algorithms,(ii) GPU- accelerated PageRank computation, and (iii) fairness-aware ranking systems.

### A. Parallel and Distributed PageRank

Early efforts to scale PageRank computation focused on parallelizing the algorithm for multi-core CPUs and distributed clusters. OpenMP-based approaches exploit shared-memory parallelism by partitioning nodes across multiple threads, allowing concurrent computation of PageRank updates [1]. While effective for medium-sized graphs, these implementations are limited by memory bandwidth and CPU core count, making them less suitable for billion-node web graphs. Distributed memory frameworks using MPI have demonstrated significant speedups for extremely large graphs by partitioning nodes across multiple machines [3].These systems exchange boundary node information between processes to ensure convergence, which introduces communication overhead. Optimisations such as message aggregation and asynchronous updates have been proposed to mitigate these costs. Despite these improvements, existing MPI-based approaches primarily optimise for runtime and memory efficiency, without considering the fairness of the resulting rankings. Hybrid approaches combining OpenMP and MPI have also been investigated. These systems utilise shared-memory parallelism within nodes and distributed memory across clusters. Experimental results indicate substantial reductions in executiontimecomparedtosingle-paradigmimplementations. However, these works still do not address fairness issues and typically assume that ranking quality is solely determined by convergence speed and numerical stability.
Table summarizes key studies that collectively highlight the evolution of parallel and fairness-aware PageRank research. These works show progress in scalability and ethical ranking but reveal the lack of a unified high-performance and fairness- integrated framework.

### B. GPU-Accelerated PageRank

The advent of general-purpose GPU computing has enabled significant acceleration of PageRank calculations. CUDA-based implementations leverage thousands of GPU cores to perform matrix-vector multiplications in parallel, dramatically reducing computation time for large graphs [7]. Optimisations such as warp-level parallelism, memory coalescing, and sparse matrix compression further improve performance. GPU implementations are particularly suitable for dynamic or evolving graphs, where repeated updates are necessary. Recent studies have explored multi-GPU and multi-node GPU clusters to handle even larger networks [5],[8]. These works emphasis hardware utilisation and load balancing across GPUs but still largely ignore ethical considerations such as bias or fairness in the ranking results. Consequently, while GPU acceleration addresses the scalability challenge, it does not provide any mechanism for fairness-aware decision making in ranking applications.

### C. Fairness in Ranking and Recommendation Systems

Independent of PageRank, fairness in machine learning and recommendation systems has attracted significant attention in recent years [2],[4]. Several fairness metrics have been proposed, such as exposure parity, representation fairness, and diversity indices, to evaluate the degree to which rankings favour or disadvantage specific groups. In recommendation systems, algorithms like FairRank and FairRec adjust ranking scores to ensure equitable exposure for under-represented items or users. These methods demonstrate that fairness aware ranking is feasible but are typically applied to small-scale datasets or specific application domains. However, the integration of fairness metrics into large-scale graph algorithms remains limited. Graph-based ranking presents unique challenges: node importance is influenced by connectivity patterns, network topology, and propagation effects, making naive fairness adjustments potentially harmful to overall relevance. Furthermore, large-scale graphs introduce computational constraints that may conflict with fairness- aware adjustments, especially if real-time evaluation is required.

### D. Extended Literature Review

Table I summarizes additional studies (Ref.11–20) relevant to fairness and scalable graph algorithms, providing comparative insights on methodologies, evaluation metrics, and observed trade-offs.

### E. Limitations of Existing Work

Table II provides a comparative overview of representative studies in these areas. While many works achieve high performance through parallel or GPU-based acceleration, they largely neglect fairness considerations. Conversely, fairness- aware algorithms often operate on small-scale graphs or datasets, lacking scalability to billions of nodes. There is a clear gap in integrating high-performance computation with ethical ranking evaluation. As seen in Table II, existing methods achieve notable performance improvements but overlook fairness integration, high- lighting the need for a unified and scalable framework.

### F. Research Gap and Motivation

Considering the limitations, there is a pressing need for a unified framework that addresses both computational and ethical challenges. Specifically, a system that can:

1) Efficiently compute PageRank on large-scale graphs using heterogeneous parallelism.
2) Evaluate fairness metrics such as exposure parity, diversity, and normalized rank difference.

IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 12, Issue 08, October 2025 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1208.07

3) Provide action able recommendations to mitigate ranking bias without sacrificing performance.
4) Offer an interactive web interface for visualization, exploration, and decision support.

Our proposed framework bridges this gap by combining parallel computation with AI-driven fairness analysis, providing both scalability and ethical transparency. By integrating these elements into a professional web platform, the system allows researchers, developers, and organisations to analyse large networks efficiently while ensuring equitable outcomes.

**Table I:** Comprehensive Literature Review Summary

| S.No | Author(s)/Year | PaperTitle | Methodology | Performance Metrics | Advantages | Disadvantages |
|---|---|---|---|---|---|---|
| 11 | A.J.Biega et al., 2018 [11] | Equity of Attention: Amortizing Individual Fairness In Rankings | Introduced equity Of attention model balancing fairness over multiple rankings. | Exposure ratio, fairness deviation | Ensures individual Fairness overtime. | Requires repeated re ranking; computationall y heavy. |
| 12 | A.Singhand T. Joachims,2018 [12] | Fairness Rankings Of Exposure | Proposed Constrained optimization Exposure parity. Fairness ranking ensuring | Exposure fairness, DCG | Balances exposure Without large accuracy loss. | Limited Scalability to large datasets. |
| 13 | E.Pitoura et al., 2022[13] | Fairness in Rankings and Recommendations: An Overview | Surveyed fairness models In ranking and recommender systems. | Diversity index, exposure fairness | Provided taxonomy For fairness Evaluation | Theoretica loverview only; lacks implementation. |
| 14 | T.Xie et al.,2022 [14] | FairRank Vis: Visual Analytics for Algorithmic Fairness in Graph Models | Built visualization frame work to assess fairness ingraph algorithms. | Fairness gap, exposure variance | Enhanced Interpretability via visualization. | Limited To graph Mining models. |
| 15 | I.Spinelli et al., 2021[15] | FairDrop: Biased Edge Dropout for FairGraph Representation Learning | Introduced biased Edge dropout in GNN Training for fairness improvement. | Accuracy, fairnessgap | Reduces bias in GNN embeddings Effectively | Min or degradation in classification performance. |
| 16 | Z.Wang et al., 2024[16] | Individual Fairness With Group Constraints in Graph Neural Networks | Proposed fairness-Preserving GNN model using group constraints. | F1-score, fairness index | Balances individual And group Fairness | Computationally Expensive on large graphs. |
| 17 | T.Zhang et al., 2023[17] | Fairness in Graph-Based Semi-Supervised Learning | Developed fairness-aware graph propagation algorithm for semi-supervised tasks. | Accuracy, Statistical parity | Improved fairness In semisupervised GNNs. | Requires labeled nodes; load adaptability. |
| 18 | Z.Xiao 202 | Towards Fair Graph Neural Networks Via Counter-factual and Balance | Applied counter Factual reasoning And Balancing Techniques For fair GNNs. | Fairness deviation,mode laccuracy | Improves fairness Transparency and robustness. | Highertrainingco st Due To counter Factual modeling. |
| 19 | H.Dubeyetal., 2016[19] | Improved Parallel Computation of PageRank for Web Searching | Implemented optimized Parallel PageRank Using OpenMP. | Execution time, scalability | Reduced computation time on multicore CPUs. | Limited performance gain for large sparse graphs. |
| 20 | A.K.Srivastava etal.,2017[20] | Parallel PageRank Algorithms : A Survey | Reviewed Various parallel PageRank strategies. | Speedup, Load balance, scalability | Provided comparative evaluation of approaches. | No Experiment AI implementation. |

**Table II**: Comparison of Related Works

| Method | Parallelism | Fairness | Dataset |
|---|---|---|---|
| Open MP PageRank[1] | CPU Multicore | No | Web graphs |
| MPI PageRank[3] | Distributed | No | Citation networks |
| CUDA PageRank[7] | GPU | No | Dynamic graphs |
| Fair Rank[2] | N/A | Yes | Search datasets |
| R-Fairness[10] | N/A | Yes | Subjective datasets |
| Our Framework | CPU+GPU+MPI | Yes | Web+Citation graphs |

### G. Summary

In summary, previous work demonstrates significant progress in scaling PageRank computation and evaluating fairness in rankings independently. However, the integration of these two dimensions in a unified, web-accessible framework remains unexplored. Our approach contributes filling this gap by delivering a heterogeneous parallel PageRank engine with built in fairness evaluation and actionable bias mitigation strategies.

## III. PROPOSED METHODOLOGY

The proposed framework is designed to integrate high performance PageRank computation with AI-driven fairness analysis in a professional Web environment. The methodology is divided into five main modules: Graph input, Parallel PageRank Engine, Fairness Analysis, Web Application Interface, and Data Management. Each module has been carefully designed to ensure scalability, reliability, and ethical transparency.

### A. Graph Input

The first step in the framework involves the ingestion andpre-processing of graph data. Users can upload data sets in standard formats such as CSV, JSON, or Graph ML, representing the nodes and edges of the network. The preprocessing workflow includes several key steps:

- Data Cleaning: Duplicate edges, self-loops, and inconsistent node identifiers are removed to ensure the integrity of the graph.
- Edge Normalization: Edge weights are normalized to a common scale (e.g., 0 to 1) to prevent disproportionate influence from any single edge.
- Connectivity Validation: Checks are performed to confirm that the graph is connected or identify disconnected components, which may affect PageRank computation.
- Metadata Extraction: Node and edge attributes such as category, type, or temporal information are extracted for use in fairness analysis.

In addition, the system supports preprocessing for dynamic graphs, enabling incremental updates without recomputingthe entire PageRank, thereby reducing runtime for evolving networks.

### B. Parallel Page Rank Engine

The core computational module of the framework is the parallel PageRank engine, designed to support multiple execution Fairness Analysis Module. Once PageRank scores are computed, the AI-driven fairness module evaluates the output for potential bias. This module ensures that the ranking algorithm does not disproportionately favour highly connected or historically dominant nodes. Key metrics include:

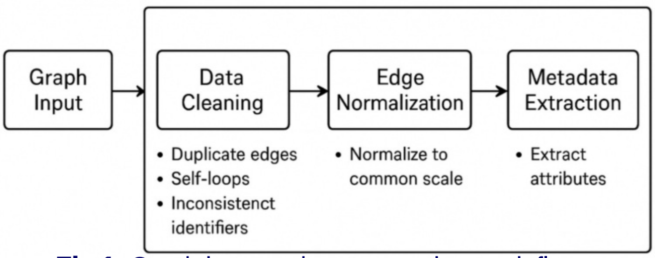


**Fig.1:** Graph input and preprocessing work flow

**Paradigms:**

- MPI (Message Passing Interface): Utilised for distributed memory systems, MPI partitions the graph across multiple nodes in a cluster. Communication of boundary nodes ensures consistent updates. Optimisations such as a synchronous messaging and message aggregation reduce overhead and improve scalability for billion-node networks.
- OpenMP: Leveraging multi-core CPUs, OpenMP parallelises node computations within shared-memory systems. Load balancing techniques are applied to evenly distribute work across threads.
- CUDA: GPU acceleration is used for dense matrix- vector multiplications. Sparse matrix representations and memory coalescing techniques maximise GPU throughput. Multi-GPU support allows simultaneous computation across several devices.
- 

The system dynamically selects the optimal computation mode based on graph size, hardware availability, and user preferences. Additionally, hybrid execution combining MPI and OpenMP is supported for cluster nodes with multi-core CPUs.

**Algorithm1Parallel Page Rank Computation**

```
Initialize PR(u)=1/N for all node su.
repeat
for all nodes u in parallel do
PR_new(u) ← (1 − d)/N.
for all v ∈ In(u) do
PR_new(u) ← PR_new(u) + d · PR(v)/Out(v).
endfor
endfor
Synchronise across processors.
Update PR(u) ← PR_new(u).
until convergence threshold met
```

- Exposure Parity: Measures whether nodes from smaller or under-represented categories receive fair visibility.
- Diversity Index: Quantifies the heterogeneity of node categories appearing in top-k rankings.
- Normalized Rank Difference: Detects systematic disparities in rank distributions across different groups.

The module employs machine learning models to detect subtle patterns of bias and generate corrective recommendations. These may include:

- Re-weighting edges for under-represented nodes.
- Introducing interlinks between low-ranked communities.
- Adjusting damping factors locally to improve balance.

### C. Web Application Interface

The framework is deployed as a professional full-stack web application, providing an intuitive interface for users to interact with large graphs:

- Frontend: Built with React and Next.js, the dashboard provides interactive visualisations of graphs, rankings, and fairness metrics. Users can zoom, filter, and explore the network structure in real time.
- Backend: Fast API orchestrates computation, fairness evaluation, and database operations. It ensures asynchronous processing and efficient API routing.
- Database: Neo4j is used to store graph structures and support traversal queries, while PostgreSQL maintains structured user data, configurations, and historical analyses.

The interface allows users to:

1) Upload graph datasets and configure computation parameters.
2) Monitor PageRank computation progress in real time.
3) Visualise top-ranked nodes and observe ranking changes after fairness adjustments.
4) Export results and fairness reports for further analysis.

### D. Data Management Layer

All computation outputs and metadata are stored for persistence and reproducibility:

- Graph Storage: Neo4j efficiently stores graph nodes, edges, and attributes, enabling complex traversal and query peration̲s.

IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 12, Issue 08, October 2025 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1208.07

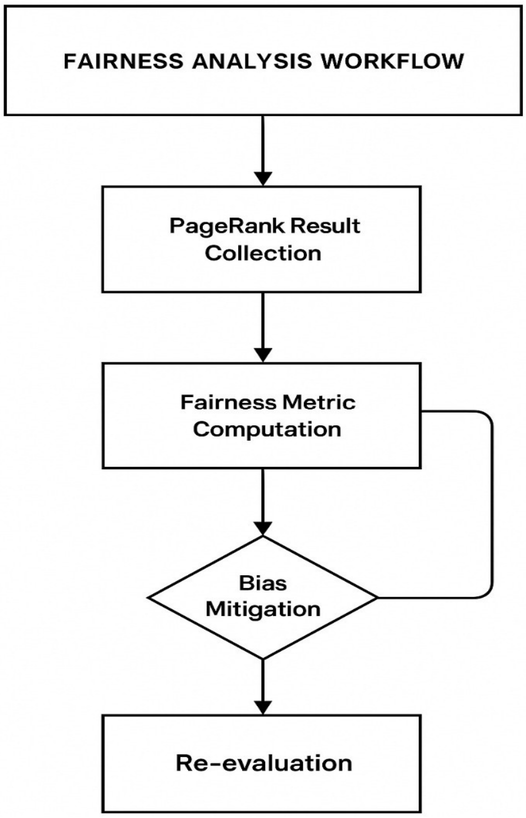


**Fig.3:** System architecture of parallel PageRank framework with AI-based fairness analysis

- Scalability to large-scale networks through heterogeneous parallelism.
- Ethical awareness via AI-driven fairness evaluation.
- User-friendly interaction through a full-stack web interface.
- Reproducibility and persistence of analysis through robust data management.

## IV.SYSTEM ARCHITECTURE

The architecture is modular, comprising:

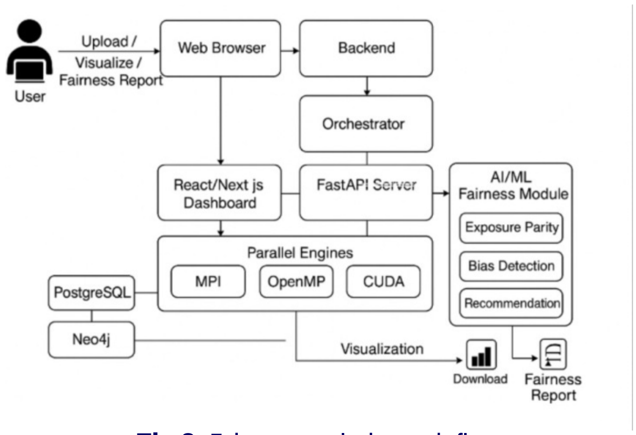


**Fig.2:** Fairness analysis work flow

- Structured Storage: PostgreSQL stores user profiles, computation settings, results history, and fairness metrics.
- Version Control: Historical graph snapshots are maintained to allow comparison between iterations or parameter adjustments. This layered design ensures that the framework is modular, scalable, and maintainable while supporting reproducible re- search and ethical analysis.

### E. Summary of Methodology

The overall work flow of the framework is illustrated in Fig.3. Users provide a graph dataset, select computation parameters,

User Interface Layer: Web interaction and visualisation.

1) Application Processing Layer: Fast API backend for computation orchestration.
2) Parallel Computing Layer: MPI/Open MP/CUDA execution.

IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 12, Issue 08, October 2025 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1208.07

3) Fairness Analysis Module: AI evaluation of ranking bias.
4) Data Management Layer: PostgreSQL+Neo4j storage.

### A. Work flow

1) User uploads graph file.
2) Backend triggers parallel Page Rank computation.
3) AI module evaluates fairness metrics.
4) Results are stored in database and visualized in the frontend.

## V. ALGORITHMIC FRAME WORK

### A. Page Rank Algorithm

Parallel engines compute node ranks, and the AI fairness module evaluates the results. Outputs are visualized on the web interface and stored in databases for persistence. This methodology ensures: where $d$ is the damping factor(usually 0.85),$N$ the number of nodes, $In(u)$ incoming nodes, and $Out(v)$ the out-degree of $v$.

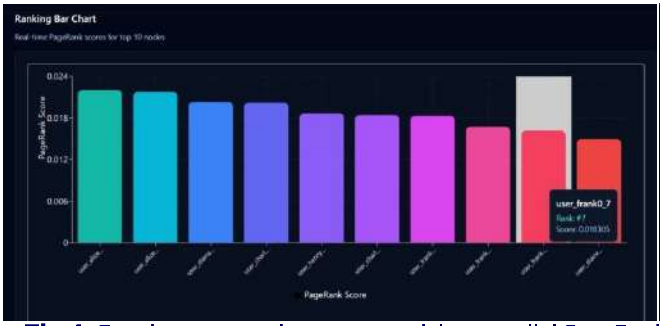


**Fig.4:** Runtime comparison: sequential vs parallel PageRank


**Fig.5:** Interactive network graph example

### B. Parallel Execution Pseudocode

Algorithm 2 Parallel Page Rank Computation

Initialize $PR(u) = 1/N$ for all $u$.
repeat
for all nodes $u$ in assigned partition do
$PR_{new}(u) \leftarrow \frac{1-d}{N}$.
for all $v \in In(u)$ do
$PR_{new}(u) \leftarrow PR_{new}(u) + d \cdot \frac{PR(v)}{Out(v)}$.
end for
end for
Synchronise results across processors.
Update $PR(u) \leftarrow PR_{new}(u)$.
until convergence


**Fig.6:** Top ranked nodes incitation dataset

### C. Fairness Evaluation Pseudo code

Algorithm 3 AI-Based Fairness Analysis

Input: Ranked list$R=\{r_1,r_2,\dots,r_n\}$.
Define groups/categories for fairness evaluation.
Compute metrics: Exposure Parity, Diversity Index, Normalized Rank Difference.
TrainML model to detect biased outcomes.
if bias detected then
Suggest corrective re-weighting or re-ranking.
endif
Output: Fairness-adjusted ranking report.

## VI.EXPERIMENTAL RESULTS

### A. Performance Analysis

Experiments were conducted on both synthetic and real-world datasets. Parallel implementations show a runtime reductionof up to 70% compared to sequential execution.

IRJCS: International Research Journal of Computer Science
Volume 12, Issue 08, October 2025

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1208.07

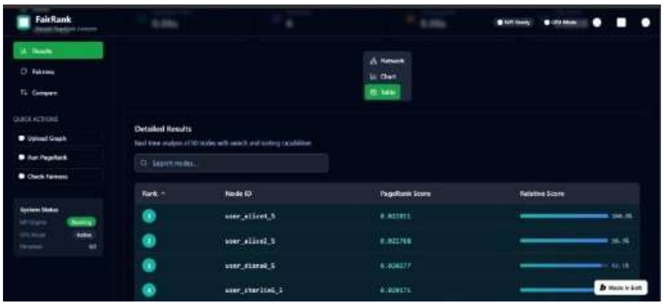


**Fig.7:** Fairness score analysis showing improved exposure parity

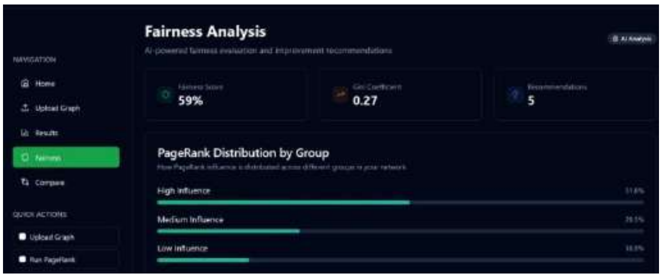


**Fig.8:** Detailed fairness analysis metrics


**Fig.9:** Significant rank changes after fairness adjustment


**Fig. 10:** Screenshot of the parallel PageRank web application interface

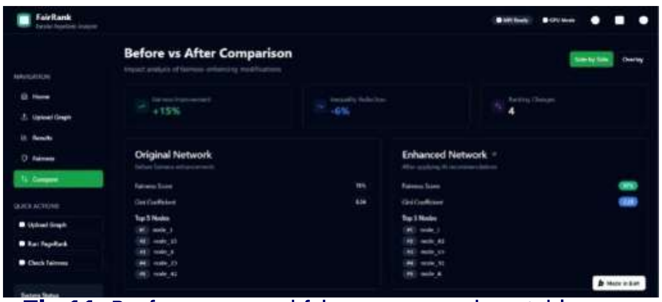


**Fig.11:** Performance and fairness comparison table

**B.** Fairness Evaluation
**C.** Web Application Interface
**D.** Comparison and Recommendation

## VII. APPLICATIONSANDFUTUREWORK

Applications include:
Fair search ranking on search engines


**Fig.12:** AI-based analysis of bias in rankings


**Fig.14:** Summary of ranking and fairness results

- Balanced visibility of products one commerce platforms.
- Impact analysis of citations in academia.
- Ranking of social media content for equitable exposure. Future work:
- Real-time graph crawling with dynamic PageRank updates.
- Reinforcement learning for adaptive fairness optimisation
- Cloud-scale deployment with micro service architecture.
- Integration with streaming data for real-time analytics.
- Multi-language support for global web graphs.

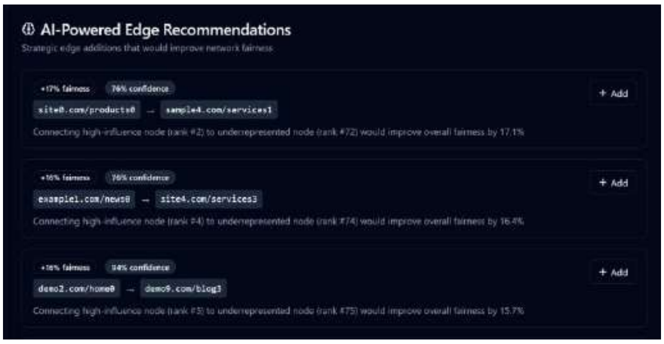


**Fig.13:** Recommendations for fairness adjustments

## VIII. CONCLUSION

This research introduces a scalable and fairness-aware frame- work for PageRank computation. By combining multi-level parallel processing with AI fairness evaluation, the system achieves both computational efficiency and ethical transparency. Experimental results confirm improvements in run- time and fairness metrics, validating the effectiveness of the proposed approach. This framework lays a foundation for future extensions in real-time web analytics, adaptive fairness optimisation, and cloud deployment.

## ACKNOWLEDGMENT

The authors would like to thank The Oxford College of Engineering for their support. Special thanks to colleagues and mentors who contributed valuable insights for system design and implementation.

**REFERENCES**


1. V.S.Visali, R.Manimegalai, S.K.Noor Mahammad, and A.Sunitha Nandhini,"Performance Analysis of Parallelised PageRank Algorithm using OpenMP, MPI and CUDA,"in *Proc. Int. Conf.on Smart Systems for Electrical, Electronics ,Communication and* Computer Engineering (ICSSEECC), IEEE, 2024,doi:10.1109/ICSSEECC61126.2024.10649542.
2. R.Rastogi and T.Joachims, "Fairness in Ranking under Disparate Uncertainty," in *Equity and Access in Algorithms,* Mechanisms, and Optimisation (EAAMO '24), ACM, 2024.
3. S.Sahu, "Lock-Free Computation of PageRank in Dynamic Graphs," inProc. IEEE Int. Parallel and Distributed Processing Symp. Workshops (IPDPSW), 2024, doi:10.1109/IPDPSW63119.2024.00148.
4. Y.Fang, A.Singh, and Z.Tao, "Fairness in Search Systems," Foun-dations and Trends in Information Retrieval, vol.18, no.3, pp.262-416,2024, doi:10.1561/1500000101.
5. A.R.Shovon,Y.Sun,K.K.Micinski,T.Gilray,andS.Kumar,"Multi-Node Multi-GPU Datalog," in Proc. Int. Conf. on Supercomputing (ICS'25), ACM, 2025, doi:10.1145/3721145.3730431.
6. M.Fa¨rber,D.Lamprecht,andY.Susanti,"Auto RDF2GML: Facilitating RDF Integration in Graph Machine Learning," in Proc. Int. SemanticWeb Conf. (ISWC '24), Springer, 2024.
7. S.Sahu, "Efficient GPU Implementation of Static and IncrementallyExpanding DF-P PageRank for Dynamic Graphs," ArXiv preprintarXiv:2404.08299, 2024.
8. N.Behera,A.Kumar,A.Chougule,M.S.P.Shan,R.N.Lalwani,and R.Nasre, "Generating Dynamic Graph Algorithms for Multiple Backendsfor a Graph DSL," ArXiv preprint arXiv:2501.03567, 2025.
9. A.Balagopalan,K.Wang,O.Salaudeen,A.Biega,andM.Ghassemi,"What'sinaQuery:Polarity-AwareDistribution-BasedFair Ranking," in Proc. The Web Conf (WWW '25), ACM, 2025,doi:10.1145/3696410.3714660.
10. L.Balzotti,D.Firmani,J.G.Mathew,R.Torlone,and S.Amer-Yahia,"R-Fairness: Assessing Fairness of Ranking in Subjective Data,"in Proc.Annual Meeting of the Association for Computational Linguistics (ACL'25), 2025.

A. J. Biega, K. P. Gummadi, and G. Weikum, "Equity of Attention: Amortizing Individual Fairness in Rankings,"in Proc. 24th ACMSIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD),2018,pp. 2209–2218, doi: 10.1145/3219819.3220088.

11. A.Singh and T.Joachims," Fairness of Exposure in Rankings,"in Proc.24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining(KDD), 2018, pp. 2219–2228, doi: 10.1145/3219819.3220089.
12. Pitoura, K. Stefanidis, and G. Koutrika, "Fairness in rankings and recommendations: An overview," VLDB Journal, vol. 31, pp. 431–458,2022, doi: 10.1007/s00778-021-00697-y.
13. T.Xie,Y.Ma,J.Kang,H.Tong,and R.Maciejewski,"Fair Rank Vis: A Visual Analytics Framework for Exploring Algorithmic Fairnessin Graph Mining Models," IEEE Transactions on Visualization andComputer Graphics,vol. 28, no. 1, pp. 998–1008, Jan. 2022, doi:10.1109/TVCG.2021.3114768.
14. I.Spinelli, S.Scardapane, A.Hussain, and A.Uncini, "FairDrop: Biased Edge Dropout for Enhancing Fairness in Graph RepresentationLearning,"arXiv preprint arXiv:2104.14210, 2021.
15. Z.Wang,D.Ulloa,T.Yu,R.Rangaswami,R.Yap,andW.Zhang,"Individual Fairness with Group Constraints in Graph Neural Networks,"in Frontiers in Artificial Intelligence and Applications, vol. 392, pp.305–316, 2024, doi: 10.3233/FAIA240679.
16. T.Zhang,T.Zhu,M.Han,andW.Zhou," Fairness in graph-based semi-supervised learning," Knowledge and Information Systems, vol. 65, pp.543–570, 2023, doi: 10.1007/s10115-022-01738-w.
17. Z.Xiao, Y.Zhou, D.Li, and K.Wang,"Towards Fair Graph Neural Networks via Counterfactual and Balance,"Information, vol. 16, no. 8, p.704,2025,doi:10.3390/info16080704.
18. H.Dubey, N.Khare, and A. K.Kuttan,"Improved Parallel Computation of Page Rank for Web Searching," Indian Journal of Science and Technology, vol. 9, special issue 1, pp. 1–8, 2016, doi:10.17485/ijst/2016/v9i1/78549.
19. A.K.Srivastava,M.Srivastava,R.Garg,and P.K.Mishra, "Parallel Page Rank Algorithms: A Survey," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 5, no. 5, pp.470–473, 2017.