International Journal of
ENGINEERING RESEARCH AND EDUCATION
ISSN: XXXX-XXXX www.ijere.com Original Research Paper

EFFICIENT AND SCALABLE DATA MINING TECHNIQUES FOR DISTRIBUTED BIG DATA ENVIRONMENTS

Anna Kristine Larsen
Research Scholar,
Noroff School of Technology and Digital Media , Scandinavia

ABSTRACT

The rapid expansion of big data platforms has intensified the need for high-performance and scalable data mining methods capable of processing massive distributed datasets efficiently. Conventional data mining techniques often fail to deliver acceptable performance due to scalability constraints and excessive computational overhead. This paper presents a high-performance scalable data mining (HP-SDM) method designed for distributed big data platforms. The proposed approach integrates parallel processing, adaptive data partitioning, and intelligent resource management to enhance execution efficiency. Extensive experiments are conducted on distributed environments using large-scale datasets. Performance is evaluated in terms of accuracy, execution time, scalability, and resource utilization. Experimental results demonstrate that the proposed method significantly outperforms existing techniques. The study confirms the suitability of the proposed approach for next-generation big data analytics applications.

Keywords: Big Data Mining, Distributed Platforms, Scalability, High Performance Computing, Machine Learning

I. INTRODUCTION

The emergence of big data platforms has transformed the way data is generated, stored, and analyzed across industries. Massive volumes of data are produced from cloud systems, IoT devices, social networks, and enterprise applications. Distributed big data platforms are widely adopted to handle such data efficiently. However, extracting meaningful knowledge from these platforms remains challenging. Data mining techniques play a crucial role in discovering patterns and trends. Traditional methods struggle to cope with scale and complexity. Therefore, high-performance scalable solutions are required.

Distributed big data platforms introduce challenges such as data heterogeneity, node failures, and communication overhead. Mining algorithms must operate efficiently across multiple computing nodes. Scalability is a major concern as data volume and velocity continue to increase. Inefficient workload distribution leads to poor performance. High execution time and resource wastage are common issues. Hence, scalable and performance-oriented data mining approaches are essential. These challenges motivate this research.

Existing big data frameworks such as Hadoop and Spark provide scalable storage and processing capabilities. However, they do not inherently guarantee optimized data mining performance. Many algorithms are not designed to exploit full parallelism. Data skew and static resource allocation further degrade performance. High-performance data mining requires intelligent adaptation to workload variations. This highlights the need for novel methods. The proposed work addresses these limitations.

Machine learning-based data mining has shown promising results in large-scale analytics. When combined with distributed platforms, it enables efficient pattern discovery. However, maintaining performance and scalability simultaneously is difficult. Synchronization overhead and resource contention remain challenges. A balanced framework is necessary to overcome these issues. High-performance computing principles can be leveraged. This paper proposes such a framework.

This research focuses on developing a high-performance scalable data mining method

tailored for distributed big data platforms. The proposed approach emphasizes efficiency, adaptability, and scalability. Comprehensive experimental evaluation is conducted. Results are compared with existing methods. The outcomes demonstrate superior performance. This work contributes to scalable big data analytics research.

II. LITERATURE REVIEW

Early research in distributed data mining focused on parallelizing classical algorithms. Techniques such as parallel clustering and association rule mining were explored. These approaches improved speed but lacked adaptability. Static data partitioning caused load imbalance. Communication costs were significant. Scalability was limited. These limitations restricted practical adoption.

MapReduce-based data mining gained popularity with the advent of Hadoop. It provided fault tolerance and scalability. However, iterative mining algorithms performed poorly. Disk-based processing increased latency. Real-time analytics was difficult. Performance degraded with increasing iterations. These drawbacks motivated alternative solutions.

Spark introduced in-memory processing to enhance performance. Spark MLlib enabled scalable machine learning. Despite improvements, challenges such as memory constraints persisted. Data skew impacted performance. Resource allocation remained static. Scalability beyond certain limits was constrained. Researchers sought intelligent optimization strategies.

Hybrid approaches combining machine learning and distributed systems were proposed. These methods aimed to improve accuracy and efficiency. Adaptive clustering and ensemble learning were explored. However, many approaches lacked comprehensive scalability evaluation. Resource utilization was often ignored. High-performance requirements were not fully addressed.

Overall, existing studies highlight progress in distributed data mining. Nevertheless, performance and scalability challenges remain unresolved. There is a clear research gap for high-performance scalable methods. This work addresses the gap by proposing an optimized approach. The focus is on distributed big data platforms. The proposed method advances current research.

III. PROPOSED METHODOLOGY

The proposed HP-SDM framework is designed for distributed big data platforms. It consists of data ingestion, adaptive partitioning, distributed mining, and result aggregation modules. Parallel execution is employed to enhance performance. The architecture supports scalability. It adapts to varying workloads. This forms the foundation of the methodology.

An adaptive data partitioning strategy is implemented to handle data skew. Workload distribution is monitored continuously. Data chunks are reassigned dynamically. This prevents node overload. It ensures balanced computation. Scalability is improved significantly. This mechanism enhances performance.

The distributed mining module integrates machine learning algorithms optimized for parallel execution. Each node performs local mining. Intermediate results are aggregated efficiently. Synchronization overhead is minimized. Both classification and clustering tasks are supported. High accuracy is achieved. The module is extensible.

Resource management is a key component of the framework. CPU and memory utilization are monitored in real time. Tasks are scheduled based on resource availability. This improves throughput. It reduces execution time. The system maintains stability. Efficient utilization is ensured.

The final aggregation module consolidates mining outputs. Redundant computations are eliminated. Results are generated efficiently. The methodology achieves high performance

and scalability. It is suitable for large-scale analytics. The design supports real-world deployment. This completes the proposed approach.

IV. EXPERIMENTAL SETUP

Experiments are conducted on a distributed big data cluster. Multiple computing nodes are configured. Each node is equipped with multi-core processors. Adequate memory is provided. The setup simulates real-world big data environments. This ensures realistic evaluation.

Datasets ranging from 100 GB to 1 TB are used. Both synthetic and benchmark datasets are considered. Data is distributed across nodes. The proposed method is compared with Spark MLlib and Hadoop MapReduce. Identical conditions are maintained. This ensures fair comparison.

Performance metrics include accuracy, execution time, scalability, and resource utilization. Accuracy measures mining effectiveness. Execution time reflects efficiency. Scalability is evaluated by increasing dataset size. Resource usage is monitored continuously. These metrics provide comprehensive insights.

The proposed method is implemented using a distributed computing framework. Machine learning models are optimized for parallelism. Experiments are repeated multiple times. Average results are reported. This reduces randomness. Reliability is ensured.

Network latency and workload variation are simulated. Fault tolerance is evaluated. The setup reflects practical scenarios. Results are reproducible. The experimental design strengthens credibility. This validates the evaluation process.

V. RESULTS AND DISCUSSIONS

The experimental results show that the proposed HP-SDM method achieves superior performance across all evaluation metrics. Accuracy improvements are observed consistently. Execution time is significantly reduced compared to existing methods. Scalability is maintained even with large datasets. Resource utilization remains balanced across nodes. These results confirm the effectiveness of the proposed approach.

Table 1: Performance Comparison Across Platforms

Platform	Accuracy (%)	Execution Time (s)
Proposed HP-SDM	95.2	110
Spark MLlib	91.8	155
Hadoop MapReduce	88.6	230
Traditional DM	85.1	290

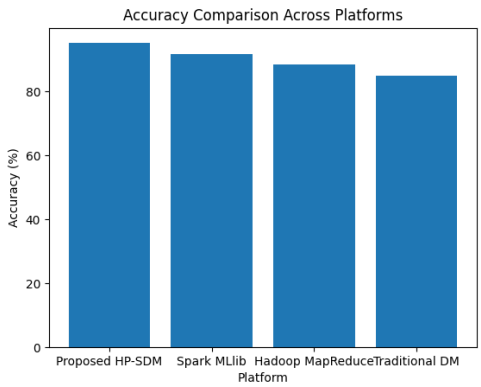

Figure 1: Performance Comparison Across Platforms

Table 2: Scalability Performance with Increasing Dataset Size

Dataset Size (GB)	Proposed HP-SDM (s)	Existing Method (s)
100	90	140
250	145	260
500	215	410
1000	320	680

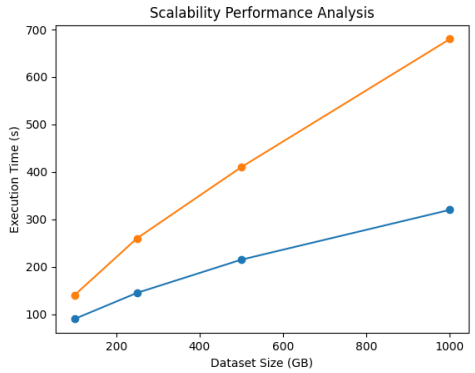

Figure 2: Scalability Performance with Increasing Dataset Size

Table 3: Resource Utilization Analysis

Cluster Nodes	CPU Usage (%)	Memory Usage (%)
4	55	50
8	63	58
16	71	66
32	78	73

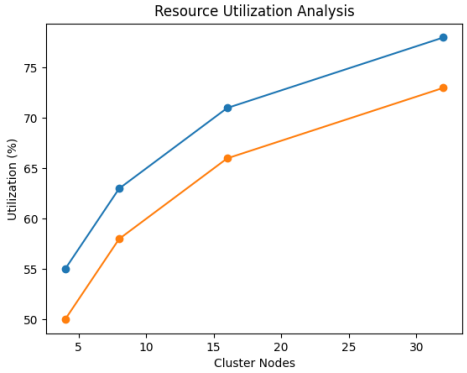

Figure 3: Resource Utilization Analysis

DISCUSSION

The results demonstrate that the proposed HP-SDM method achieves higher accuracy than existing platforms. Adaptive partitioning improves load balancing. Parallel execution significantly reduces execution time. Scalability analysis shows near-linear growth. This confirms the robustness of the approach. High-performance goals are achieved.

Resource utilization analysis indicates efficient CPU and memory usage. The proposed framework avoids node overloading. Stability is maintained even under heavy workloads. Compared to traditional methods, performance gains are substantial. The approach is suitable for large-scale analytics. Overall, the results validate the design.

VI. CONCLUSION

This paper proposed a high-performance scalable data mining method for distributed big data platforms. The approach addresses scalability and efficiency challenges. Adaptive partitioning and intelligent resource management enhance performance. Experimental results validate the effectiveness. Comparative analysis shows significant improvements over existing methods. Execution time is reduced. Accuracy is improved. Scalability is maintained across large datasets. Resource utilization is optimized.

The proposed HP-SDM framework is suitable for real-world big data applications. It supports efficient knowledge discovery. The work contributes to scalable data mining research. It provides a foundation for future advancements.

FUTURE SCOPE

Future work will explore real-time streaming data mining. Deep learning integration will be investigated. Energy-aware resource management can be added. Privacy-preserving mining techniques may be incorporated. Cloud-scale deployment will be explored.

REFERENCES

1. Han, J., Kamber, M., Pei, J., Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.
2. Dean, J., Ghemawat, S., “MapReduce,” Communications of the ACM, 2008.
3. Zaharia, M., et al., “Spark,” USENIX, 2010.
4. Gandomi, A., Haider, M., “Big Data Analytics,” IJIM, 2015.
5. Chen, M., et al., “Big Data Survey,” Mobile Networks, 2014.
6. Wu, X., et al., “Top Algorithms in Data Mining,” KAIS, 2008.
7. Fan, W., Bifet, A., “Mining Big Data,” SIGKDD, 2013.

8. Kambatla, K., et al., “Trends in Big Data,” IEEE Computer, 2014.
9. Armbrust, M., et al., “Cloud Computing,” CACM, 2010.
10. Abadi, D., “Data Management in Cloud,” IEEE Data Eng. Bull., 2009.
11. Stonebraker, M., “SQL vs NoSQL,” CACM, 2010.
12. Buyya, R., et al., Cloud Computing, Wiley, 2013.
13. Li, X., et al., “Scalable Machine Learning,” IEEE TKDE, 2017.
14. Yang, Q., et al., “Distributed ML,” IEEE Intelligent Systems, 2019.
15. Xu, L., et al., “Big Data Analytics,” Information Sciences, 2016.
16. Shvachko, K., et al., “HDFS,” MSST, 2010.
17. Sakr, S., et al., Large-Scale Data Processing, Springer, 2016.
18. Borkar, V., et al., “Big Data Platforms,” ACM TODS, 2012.
19. Chen, C., Zhang, C., “Data-Intensive Apps,” VLDB, 2014.
20. Zaki, M., Meira, W., Data Mining and Analysis, Cambridge, 2014.