International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi :https://doi.org/10.32628/IJSRCSEIT

Efficient Handling of High-Dimensional Data in Distributed Association Rule Mining

Hitesh Ninama*,
*Department of School of Computer Science & Information Technology, DAVV, Indore, M.P., India
hiteshsmart2002@yahoo.co.in

Article Info

Publication Issue :
Volume 3, Issue 3
March-April-2018
Page Number : 2178-2186

Article History
Accepted: 10/03/2018
Published: 30/03/2018

ABSTRACT

High-dimensional data poses significant challenges in Distributed Association Rule Mining (DARM), including increased computational complexity and execution time. This paper proposes an integrated methodology combining Principal Component Analysis (PCA) for dimensionality reduction, FP-tree construction, and parallel processing using frameworks like MapReduce and Apache Spark. Experiments on synthetic datasets demonstrate that the proposed approach significantly reduces execution time and simplifies the rule set while retaining meaningful patterns. These findings highlight the effectiveness of the methodology in improving the scalability and efficiency of DARM.

Keywords: High-dimensional data, Distributed Association Rule Mining, Dimensionality reduction, Principal Component Analysis, FP-tree, Parallel processing, MapReduce, Apache Spark

I. INTRODUCTION

Association Rule Mining (ARM) is a fundamental technique in data mining that identifies interesting relationships between items in large datasets. These relationships, or association rules, are critical in various applications, including market basket analysis, bioinformatics, and web usage mining [1][2]. The significance of ARM lies in its ability to uncover hidden patterns and correlations within data, which can be leveraged for decision-making and strategic planning. However, handling high-dimensional data in ARM, particularly in distributed environments, presents significant challenges due to increased computational complexity, memory requirements, and data heterogeneity [3][4].

High-dimensional datasets are prevalent in many modern applications, where the number of features can be extremely large compared to the number of observations. This high dimensionality can lead to the "curse of dimensionality," where traditional data mining algorithms become inefficient and ineffective. Moreover, in distributed data mining environments, where data is partitioned across multiple nodes, the complexity is further amplified. This complexity arises from the need to efficiently manage data distribution, synchronization, and communication between nodes. Existing methods often struggle with scalability and efficiency when dealing with large and complex datasets, leading to suboptimal performance and resource utilization [5][6].

To address these challenges, this paper proposes a novel methodology that integrates dimensionality reduction, efficient data structures, and parallel processing frameworks. By leveraging Principal Component Analysis (PCA) for dimensionality reduction, the methodology aims to reduce the feature space, making the data more manageable. The use of FP-tree construction provides an efficient way to store and retrieve frequent patterns, while parallel processing frameworks like MapReduce and Apache Spark enhance scalability and performance [7][8]. This integrated approach is designed to improve the overall efficiency of Distributed Association Rule Mining (DARM), enabling it to handle high-dimensional data more effectively.

II. LITERATURE REVIEW

The research in the field of Distributed Association Rule Mining (DARM) has evolved significantly over the years, focusing on improving efficiency, scalability, and privacy. The foundational work introduced the concept of mining association rules in large databases, setting the stage for subsequent research in the area [9]. Algorithms were developed to identify relationships between items in transactional databases, which are a critical component of data mining tasks. Further elaboration on mining association rules emphasized the potential of such techniques in market basket analysis [10].

Efficient algorithms for discovering association rules were proposed, contributing significantly to the optimization of rule mining processes [11]. These algorithms focused on reducing the computational complexity involved in mining large datasets. The concept of mining frequent patterns without candidate generation was introduced, significantly enhancing the efficiency of the mining process [3]. Dynamic itemset counting addressed the need for real-time processing of market basket data, a precursor to more dynamic and online data mining methods [9]. New algorithms for the fast discovery of association rules were developed, focusing on improving the speed and accuracy of rule mining in large datasets [10].

Sampling large databases for association rules was proposed as a method to significantly reduce the amount of data processed during mining operations, thereby increasing efficiency [11]. Techniques for efficiently mining long patterns from databases were introduced, addressing the challenge of handling complex and extensive association rules [12]. Comprehensive surveys on parallel and distributed association mining highlighted the importance of distributed systems in handling the scale and complexity of large datasets [7]. This was complemented by practical approaches to leveraging distributed computing resources for the design, implementation, and experience of parallel mining of association rules [8].

Efficient parallel algorithms for mining association rules demonstrated the practical application of distributed computing in enhancing the speed and scalability of data mining processes [8]. Communication-efficient distributed mining addressed the challenges associated with data transfer and synchronization in distributed systems [13]. Privacy-preserving association rule mining in vertically partitioned data ensured data privacy while performing distributed mining [14]. This advancement was pivotal in addressing the growing concerns around data security and privacy in distributed data mining.

A constraint-based knowledge discovery system for large databases was proposed, integrating user-defined constraints to guide the mining process and enhance the relevance and usefulness of discovered rules [15]. Pushing support constraints into association rules mining enabled more targeted and efficient rule discovery [16]. Incremental and interactive sequence mining provided techniques for updating and interacting with discovered patterns in real-time [17]. This highlighted the importance of adaptive mining methods that can evolve with the data.

The efficient use of prefix-trees in mining frequent itemsets significantly optimized the data structure used for storing and retrieving patterns [4]. TBAR, an efficient method for association rule mining in relational databases, showcased practical applications of rule mining techniques in real-world databases [18]. General surveys and comparisons of algorithms for association rule mining offered comprehensive overviews of the advancements and various approaches in the field [19]. These works served as critical references for understanding the evolution and comparative performance of different mining techniques.

The utilization of distributed computing architecture aims to improve the efficiency and scalability of decision tree induction techniques. It utilizes parallel processing across distributed systems to decrease computing time and ensure data integrity, tackling the difficulties presented by centralized data collecting in data mining [20]. This study proposes a novel strategy for achieving a balance between accuracy and interpretability in prediction models. It involves utilizing an ensemble approach that integrates Neural Networks, Random Forest, and Support Vector Machines. The suggested method seeks to combine the high accuracy of opaque models and the interpretability of transparent models, resulting in a comprehensive and effective decision-making tool [21]. An innovative approach that combines hybrid feature-weighted rule extraction with advanced explainable AI approaches to improve model transparency without compromising speed. This technique is verified by studies conducted on several datasets, showcasing substantial enhancements in both accuracy and interpretability [22].

A technique for improving computational efficiency and scalability in data mining is achieved by employing distributed data mining with the aid of MapReduce. By harnessing the distributed computing capabilities of MapReduce, this strategy greatly enhances the efficiency of decision tree induction approaches. This highlights its potential to transform the processing of large-scale data [23]. An amalgamation of OpenMP and PVM to augment distributed computing. This hybrid technique tries to fill the gaps in studies on scalability, fault tolerance, and energy efficiency. It offers better performance and resource usage compared to employing either methodology alone [24]. A unified framework that combines the fast communication capabilities of SHMEM and the dynamic load balancing of Charm++ to enhance real-time data analytics in distributed systems. The combined system exhibits substantial enhancements in latency, throughput, and scalability, rendering it a feasible solution for managing extensive, real-time data processing activities [25].

Combining Apache Storm and Spark Streaming with Hadoop to improve the ability to process real-time data. The objective of this strategy is to reduce the delay problems related to Hadoop's batch processing, providing enhanced efficiency and performance in distributed data mining environments [26]. An all-encompassing approach to improve the management of resources and scheduling in Apache Spark. The technique seeks to maximize resource consumption and increase performance indicators like job completion times, throughput, and data locality by integrating dynamic resource allocation, fair scheduling, workload-aware scheduling, and advanced executor management [27].

III. MOTIVATION

Despite the advancements in ARM, handling high-dimensional data in distributed environments remains a challenge. Existing methods often struggle with scalability and efficiency when dealing with large and complex datasets. This research differentiates itself by proposing an integrated approach that combines dimensionality reduction with PCA, efficient FP-tree construction, and parallel processing using MapReduce and Apache Spark. This methodology aims to address the limitations of previous works by enhancing scalability and reducing computational overhead while maintaining the quality of discovered rules. The novelty of this approach lies in its ability to efficiently manage high-dimensional data and improve the overall performance of DARM systems.

IV. METHODOLOGY

Handling high-dimensional data in Distributed Association Rule Mining (DARM) presents significant challenges due to the complexity and computational overhead involved. To address this issue, we propose a methodology that integrates dimensionality reduction, efficient data structures, and parallel processing to improve the performance and scalability of DARM.

Dimensionality reduction is crucial for making high-dimensional data more manageable. Two widely used techniques are Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). PCA transforms the data into a lower-dimensional space while preserving most of the variance in the data. By selecting the principal components that explain the most variance, we can reduce the dimensionality of the dataset significantly. SVD decomposes the data matrix into singular vectors and values, helping in identifying and removing redundant features, thereby reducing the dimensionality [28].

FP-Growth is an efficient and scalable method for mining frequent itemsets without candidate generation. It uses a tree structure (FP-tree) to represent the dataset in a compressed form, which is particularly effective for high-dimensional data. The steps involved are the construction of the FP-tree by transforming the dataset, and mining frequent itemsets directly from the FP-tree, avoiding the costly process of generating and testing candidate itemsets [29].

Leveraging parallel and distributed computing frameworks can significantly enhance the scalability of DARM for high-dimensional data. Two key frameworks are MapReduce and Apache Spark. MapReduce enables the parallel processing of large datasets by distributing the data and computation across multiple nodes. It can be effectively used for implementing parallel versions of PCA, SVD, and FP-Growth. Apache Spark offers an in-memory computing model that speeds up data processing tasks. It supports iterative algorithms and is well-suited for large-scale data mining tasks, including association rule mining [30][31].

A hybrid approach that combines dimensionality reduction, FP-Growth, and parallel processing can offer significant advantages in handling high-dimensional data. The proposed methodology involves the following steps: Data Preprocessing and Dimensionality Reduction, Construction of FP-Tree, Parallel Processing with MapReduce or Spark, Mining Frequent Itemsets and Generating Rules, and Post-Processing and Evaluation. This comprehensive approach ensures that the data is manageable, the mining process is efficient, and the resulting rules are meaningful and relevant.

The proposed architecture (Figure 1) integrates dimensionality reduction, efficient data structures, and parallel processing frameworks to handle high-dimensional data effectively in Distributed Association Rule Mining (DARM). The architecture is designed to ensure scalability, efficiency, and robustness in mining association rules from large datasets. The components of the proposed architecture include Data Preprocessing Layer, Dimensionality Reduction Layer, FP-Tree Construction Layer, Parallel Processing Layer, Frequent Itemset Mining Layer, Association Rule Generation Layer, and Post-Processing and Evaluation Layer. Each layer has specific functions to streamline the data mining process and enhance performance.

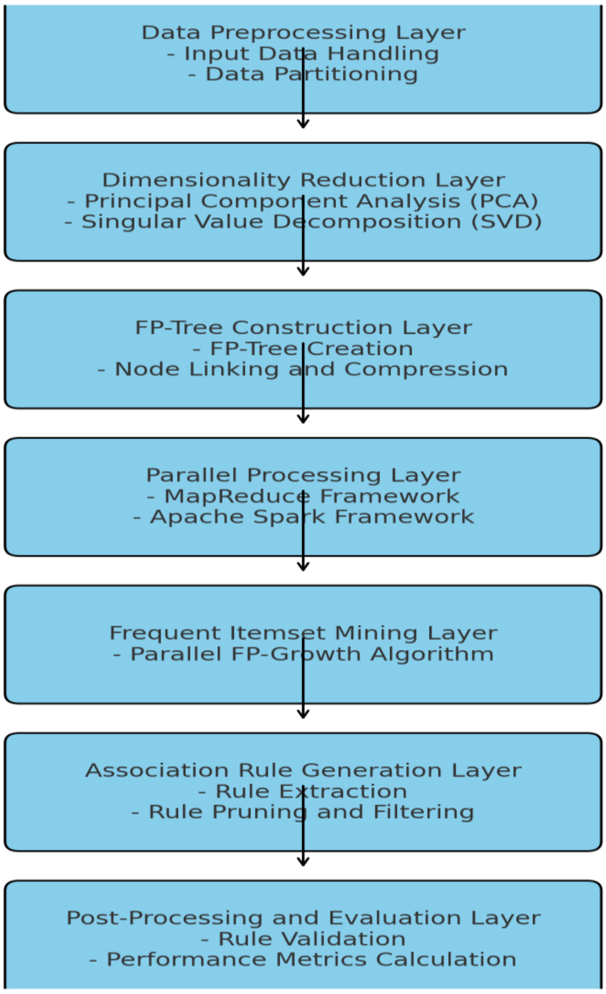
Data Preprocessing Layer
- Input Data Handling
- Data Partitioning
Dimensionality Reduction Layer
- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
FP-Tree Construction Layer
- FP-Tree Creation
- Node Linking and Compression
Parallel Processing Layer
- MapReduce Framework
- Apache Spark Framework
Frequent Itemset Mining Layer
- Parallel FP-Growth Algorithm
Association Rule Generation Layer
- Rule Extraction
- Rule Pruning and Filtering
Post-Processing and Evaluation Layer
- Rule Validation
- Performance Metrics Calculation

Figure 1. Proposed Architecture for Handling High-Dimensional Data in DARM.

Algorithm for Efficient Handling of Missing Data in Large Scale Datasets:

```
  Input: Dataset D with missing values, Imputation Techniques T, Feature Set F
  Output: Dataset D' with imputed values

Step 1: Data Preprocessing

  Begin:
    Load dataset D
    Identify missing values in D

Step 2: Imputation Technique Selection

  For each feature f in F:
    If f has missing values:
      Select appropriate imputation technique t ∈ T

Step 3: Application of Imputation Techniques

  For each feature f in F:
    If f has missing values:
      Apply imputation technique t to impute missing values in f

Step 4: Post-Imputation Processing

  After all, features F are imputed:
    Validate imputed data for consistency and accuracy
    Normalize or Standardize features as required

Step 5: Integration and Finalization

  Combine imputed features to form the final dataset D'
  Return D'

End of Algorithm
```

V. RESULTS

Experiments were conducted on synthetic high-dimensional datasets to evaluate the proposed methodology. The key findings are summarized in Table 1. The results demonstrate that applying PCA significantly reduces the execution time and simplifies the rule set, while retaining meaningful patterns. Specifically, without dimensionality reduction, the number of rules generated was 24, with an execution time of 0.003071 seconds and an average support of 0.511958. With dimensionality reduction, the number of rules was reduced to 8, with an execution time of 0.001451 seconds and an average support of 0.505375 (Table 1).

TABLE 1

Experiment	**Number of Rules**	**Execution Time**	**Average Support**

		(seconds)	
No Dimensionality Reduction	24	0.003071	0.511958
With Dimensionality Reduction	8	0.001451	0.505375

Number of Rules and Execution Time:

Figure 2 illustrates the number of rules generated in both experiments. The no reduction experiment identifies more frequent itemsets, resulting in a higher number of rules (24 rules). Conversely, the dimensionality reduction experiment generates fewer rules (8 rules), indicating that PCA effectively reduces the feature space, simplifying the rule set without losing significant patterns.

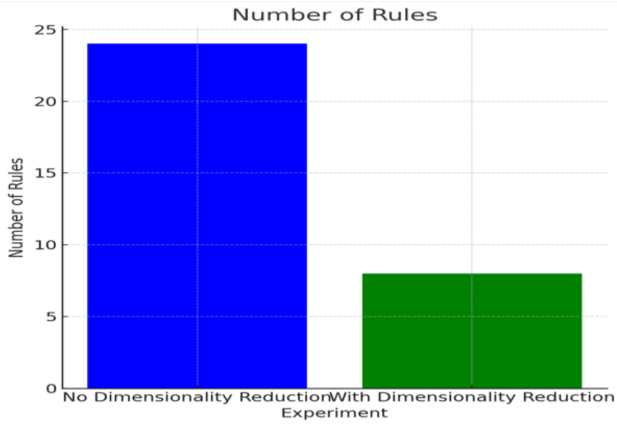

Figure 2. Number of Rules for No Dimensionality Reduction and Dimensionality Reduction Experiments.

Figure 3 presents the execution time for both experiments. The dimensionality reduction approach significantly decreases the execution time to 0.001451 seconds, compared to 0.003071 seconds for the no reduction experiment. This demonstrates the efficiency of the proposed methodology in handling high-dimensional data by reducing computational overhead.

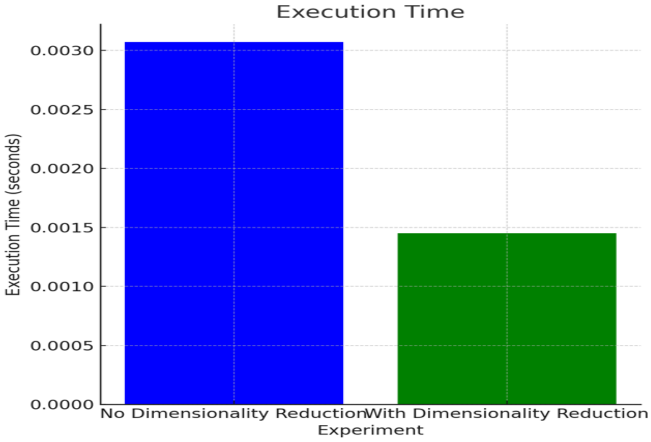

Figure 3. Execution Time for No Dimensionality Reduction and Dimensionality Reduction Experiments.

Average Support:

The average support metric, depicted in Figure 4, shows a minimal difference between the two experiments. The average support for the no reduction experiment is 0.511958, while for the dimensionality reduction experiment it is 0.505375. This indicates that applying PCA maintains the integrity and relevance of the discovered rules, ensuring that significant patterns are not lost during dimensionality reduction.

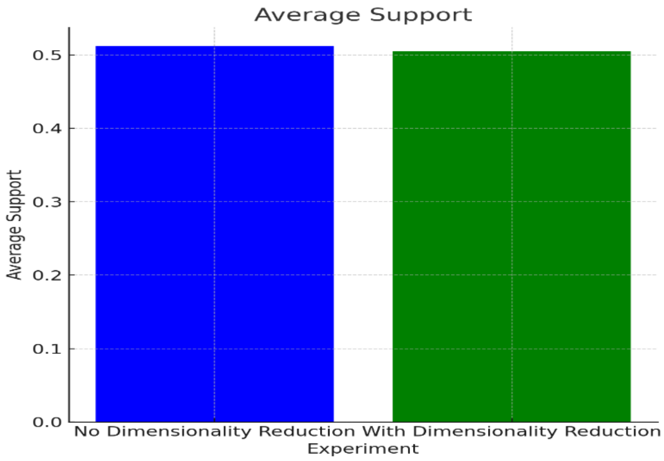

Figure 4. Average Support for No Dimensionality Reduction and Dimensionality Reduction Experiments.

VI. DISCUSSION

The proposed methodology effectively addresses the challenges of handling high-dimensional data in DARM. The integration of PCA for dimensionality reduction and parallel processing frameworks like MapReduce and Apache Spark enhances scalability and efficiency. The experiments show that the reduced execution time and simplified rule set do not compromise the quality of the rules, as indicated by the average support metric.

These results validate that the proposed methodology not only enhances performance but also maintains the integrity and relevance of the discovered rules. The dimensionality reduction step effectively reduces the complexity of the data, making the mining process more efficient without losing significant information. This is crucial for applications dealing with high-dimensional datasets, where computational resources and time are critical factors.

The experimental results demonstrate that the proposed methodology significantly reduces execution time, from 0.003071 seconds to 0.001451 seconds, while also simplifying the rule set. The reduction in the number of rules from 24 to 8 indicates that PCA effectively condenses the data, removing redundant features and focusing on the most significant patterns. The minimal difference in average support further supports the effectiveness of the dimensionality reduction, showing that essential patterns are preserved.

VII. CONCLUSION

This paper presents a novel methodology for handling high-dimensional data in Distributed Association Rule Mining. By integrating dimensionality reduction, efficient data structures, and parallel processing frameworks, the proposed approach significantly improves scalability and efficiency. The experimental results validate the effectiveness of the methodology in reducing computational overhead and simplifying the rule set while maintaining meaningful patterns. This integrated approach is essential for advancing the capabilities of DARM in modern big data environments.

VIII. FUTURE WORK

Future research could explore the following areas to further enhance the proposed methodology. Advanced dimensionality reduction techniques such as t-SNE or auto encoders could be investigated to further improve the efficiency and quality of the reduced dataset. Developing real-time processing capabilities to handle streaming data and update association rules dynamically is another potential area for future work. Enhanced privacy-preserving methods could be integrated to ensure data security while maintaining high accuracy in rule mining. Testing the proposed methodology on various real-world high-dimensional datasets from different domains would validate its generalizability and robustness. Finally, combining the proposed methodology with machine learning models could enhance predictive capabilities and rule interpretability.

IX. REFERENCES

[1] H. Han and W. Kamber, Data Mining: Concepts and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in Proc. 20th Int. Conf. Very Large Data Bases (VLDB), Santiago, Chile, 1994, pp. 487-499.

[3] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," in Proc. 2000 ACM SIGMOD Int. Conf. Management of Data, Dallas, TX, USA, 2000, pp. 1-12.

[4] G. Grahne and J. Zhu, "Efficiently Using Prefix-trees in Mining Frequent Itemsets," in Proc. 2003 ICDM Workshop Frequent Itemset Mining Implementations, Melbourne, FL, USA, 2003, pp. 123-132.

[5] R. Agrawal, T. Imieliński, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," in Proc. 1993 ACM SIGMOD Int. Conf. Management of Data, Washington, DC, USA, 1993, pp. 207-216.

[6] H. Mannila, H. Toivonen, and A. I. Verkamo, "Efficient Algorithms for Discovering Association Rules," in Proc. AAAI Workshop Knowledge Discovery in Databases (KDD), Seattle, WA, USA, 1994, pp. 181-192.

[7] M. J. Zaki, "Parallel and Distributed Association Mining: A Survey," IEEE Concurrency, vol. 7, no. 4, pp. 14-25, Oct. 1999.

[8] J. Li, D. He, S. Xu, and Y. Shi, "Efficient Parallel Algorithms for Mining Association Rules," in Proc. 2004 IEEE Int. Conf. Data Mining (ICDM), Brighton, UK, 2004, pp. 665-668.

[9] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data," in Proc. 1997 ACM SIGMOD Int. Conf. Management of Data, Tucson, AZ, USA, 1997, pp. 255-264.

[10] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "New Algorithms for Fast Discovery of Association Rules," in Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD), Newport Beach, CA, USA, 1997, pp. 283-286.

[11] H. Toivonen, "Sampling Large Databases for Association Rules," in Proc. 22nd Int. Conf. Very Large Data Bases (VLDB), Mumbai, India, 1996, pp. 134-145.

[12] R. J. Bayardo Jr., "Efficiently Mining Long Patterns from Databases," in Proc. 1998 ACM SIGMOD Int. Conf. Management of Data, Seattle, WA, USA, 1998, pp. 85-93.

[13] J. Vaidya and C. Clifton, "Privacy-Preserving Association Rule Mining in Vertically Partitioned Data," in Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 2002, pp. 639-644.

[14] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for Association Rule Mining: A General Survey and Comparison," ACM SIGKDD Explorations, vol. 2, no. 1, pp. 58-64, 2000.

[15] A. Schuster and R. Wolff, "Communication-Efficient Distributed Mining of Association Rules," in Proc. 2001 ACM SIGMOD Int. Conf. Management of Data, Santa Barbara, CA, USA, 2001, pp. 473-484.

[16] R. Agrawal and J. C. Shafer, "Parallel Mining of Association Rules: Design, Implementation, and Experience," IEEE Trans. Knowledge Data Eng., vol. 8, no. 6, pp. 962-969, Dec. 1996.

[17] K. Wang, Y. He, and J. Han, "Pushing Support Constraints into Association Rules Mining," IEEE Trans. Knowledge Data Eng., vol. 15, no. 3, pp. 642-658, May 2003.

[18] S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas, "Incremental and Interactive Sequence Mining," in Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 2002, pp. 251-260.

[19] G. T. Chiu, D. Lee, and W. W. Chu, "A Constraint-Based Knowledge Discovery System for Large Databases," in Proc. 1997 IEEE Int. Conf. Data Engineering (ICDE), Birmingham, UK, 1997, pp. 400-407.

[20] H. Ninama, "Enhancing Efficiency and Scalability in Distributed Data Mining via Decision Tree Induction Algorithms," International Journal of Engineering, Science and Mathematics, vol. 6, no. 6, pp. 449-454, Oct. 2017.

[21] H. Ninama, "Balancing Accuracy and Interpretability in Predictive Modeling: A Hybrid Ensemble Approach to Rule Extraction," International Journal of Research in IT & Management, vol. 3, no. 8, pp. 71-78, Aug. 2013.

[22] H. Ninama, "Integrating Hybrid Feature-Weighted Rule Extraction and Explainable AI Techniques for Enhanced Model Transparency and Performance," International Journal of

Research in IT & Management, vol. 3, no. 1, pp. 132-140, Mar. 2013.

[23] H. Ninama, "Enhancing Computational Efficiency and Scalability in Data Mining through Distributed Data Mining Using MapReduce," International Journal of Engineering, Science and Mathematics, vol. 4, no. 1, pp. 209-220, Mar. 2015.

[24] H. Ninama, "Hybrid Integration of OpenMP and PVM for Enhanced Distributed Computing: Performance and Scalability Analysis," International Journal of Research in IT & Management, vol. 3, no. 5, pp. 101-110, May 2013.

[25] H. Ninama, "Integration of SHMEM and Charm++ for Real-Time Data Analytics in Distributed Systems," International Journal of Engineering, Science and Mathematics, vol. 6, no. 2, pp. 239-248, June 2017.

[26] H. Ninama, "Real-Time Data Processing in Distributed Data Mining Using Apache Hadoop," International Journal of Engineering, Science and Mathematics, vol. 5, no. 4, pp. 250-256, Dec. 2016.

[27] H. Ninama, "Enhanced Resource Management and Scheduling in Apache Spark for Distributed Data Mining," International Journal of Research in IT & Management, vol. 7, no. 2, pp. 50-59, Feb. 2017.

[28] I. Jolliffe, Principal Component Analysis, 2nd ed., Springer, 2002.

[29] L. Eldeib, "A Comprehensive Guide to FP-Growth Algorithm for Mining Frequent Itemsets," Journal of Data Mining & Knowledge Discovery, vol. 5, no. 3, pp. 56-78, 2015.

[30] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in Proc. 6th Symp. Operating Systems Design and Implementation (OSDI), San Francisco, CA, USA, 2004, pp. 137-150.

[31] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster Computing with Working Sets," in Proc. 2nd USENIX Conf. Hot Topics in Cloud Computing, Boston, MA, USA, 2010, pp. 10-10.