IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AN AGENT BASED APPROACH ON USER PRIORITIES FOR THREE PHASE INTELLIGENT RECOMMENDATION AND CLOUD SERVICE NEGOTIATION

[1]Dr.B.Selva Priya [2]Aare.Lahari [3]Adepu.Neeraj [4]T.Ganesh Kumar Reddy [5]T.Nivesh

[1]Assistant professor, School of Computing, Department of Computer Science and Engineering , Bharath Institute of Higher Education And Research, Chennai, India- 600073 .

[2, 3,4,5] Student , , School of Computing, Department of Computer Science and Engineering, Bharath Institute of Higher Education And Research, Chennai, India- 600073.

Abst

In light of the rapid advancement of information technology in recent years, data has proliferated across various sources, including sensor data, social media posts, images, and raw unstructured data. This inundation of information presents a significant challenge for current data management systems, particularly in handling large volumes of unstructured data, commonly referred to as Big Data. In this study, we delve into the fundamental concepts and frameworks of Big Data tools, algorithms, and methodologies. Our focus lies in comparing traditional data mining algorithms with those specifically designed for Big Data processing, utilizing CSP/MapReduce as the fundamental scalable algorithmic framework. To empirically evaluate the performance, we implemented K…

Keywords:

Big Data, unstructured data, CSP/MapReduce, data mining algorithms, K-means, A-priori, NoSQL databases, MongoDB, HDFS, performance analysis**.**

I. INTRODUCTION

n today's dynamic cloud services market, users face the daunting task of selecting the most appropriate solutions tailored to their diverse requirements. To navigate this complexity, we introduce an innovative Agent and Approach based on User-Priorities for Three-Phase Intelligent Recommendation and Cloud Service Negotiation. This strategy harnesses the power of intelligent recommendation and negotiation agents to thoroughly analyze user preferences and system requirements. Through three sequential phases—user profiling, personalized recommendation, and dynamic satisfaction levels and optimize service provisioning efficacy amidst the myriad challenges and risks prevalent in the digital landscape.

II. LITERATURE REVIEW

[1] A comprehensive analysis identified five key challenges encountered in healthcare data management systems. These include: 1. Handling sensitive data securely. 2. Analyzing complex and heterogeneous data spaces, incorporating contextual information. 3. Managing distributed data while adhering to security and performance requirements. 4. Utilizing specialized analytics to integrate bioinformatics and systems biology data with clinical observations across various scales. 5. Implementing specialized analytics to establish the "physiological envelope" throughout the daily lives of individual patients.

[2] Explored the concept of a learning health system, as proposed by the Institute of Medicine, where the boundaries between research and clinical practice are blurred. The historical background of this concept is traced through examinations of similar initiatives in the business domain, such as knowledge management, business process reengineering, and enterprise resource planning.

[3] Highlighted the necessity of standardization to achieve interoperability for pathology test requesting and reporting. Interoperability, defined as the capability of two parties, human or machine, to exchange data or information while maintaining shared meaning, is crucial in healthcare settings.

[4] Pointed out the predominant focus of clinical research on resource-intensive causal inference, contrasting it with the largely unexplored potential of predictive analytics driven by the abundance of big data sources. The author emphasized that basic prediction, without the complexity of causal inference, becomes more feasible with the availability of big data.

[5] Envisioned the emergence of a healthcare-centric democracy and projected a surge in the volume and speed of patient-generated data. This development is anticipated to significantly influence the integration of digital health records across various platforms and their accessibility to healthcare practitioners for diagnosis and treatment purposes.

III. PROPOSED METHODOLOGY

In this section, we delve into the detailed explanation of two prominent data mining algorithms: A-priori and K-means.

A-PRIORI ALGORITHM

The A-priori algorithm is widely utilized in data mining to identify frequent item-sets within transactional databases. Its primary function is to uncover associations between items by analyzing their frequency of occurrence. For instance, consider a retail store with a transactional database (D) aiming to understand customer purchasing patterns for strategic marketing purposes. Formally, the problem can be defined as follows:

Let I = {I1, I2 … In} denote the item-set, and D represent the transactional database.
Suppose A and B are sets within a transaction T (A,B ⊑ T).
A⇒B represents a rule, where A⊑I, B⊑!, A ≠ ∅, B ≠ ∅, and A ∩ B ≠ ∅
.A⇒B rule holds with minimum support and confidence.
Support (S) is calculated as P(A ∪ B).
Confidence (C) is determined as the conditional probability P(A | B) = (Sup_Count (A ∪ B))/(Sup_Count (A)).
Originally developed at IBM by Agrawal, the A-priori algorithm functions as follows:

Scan the entire database to determine item counts (1-itemset or L1).
Join L1 with itself to generate the next frequent item-set (k-itemset), where two frequent items are considered joinable if their (k-1) subsets match.
Iterate the algorithm until the frequent item-set (Lk) becomes empty.

IV. SYSTEM ARCHITECTURE

OBJECTIVES

The research aims to achieve the following objectives:

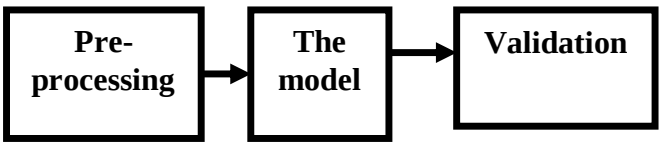
Investigate Big Data technologies, tools, and concepts to understand their implications.
Explore the paradigm shift in databases with the emergence of NoSQL databases, particularly focusing on document-oriented databases like MongoDB.
Implement common data mining algorithms, specifically the A-priori algorithm and K-means clustering algorithm, using the MapReduce model. Subsequently, compare their performance between HDFS (CSP Distributed File System) and MongoDB data stores.
BASIC DESIGN

Data Mining, its Relationship with Big Data, and its Significance Today:

Data mining is an interdisciplinary field that draws upon machine learning, artificial intelligence, and mathematical statistics to uncover and extract patterns from datasets. It goes beyond the capabilities of traditional SQL (Structured Query Language) queries. In this section, we introduce and motivate readers to delve deeper into the concept of data mining.

Data mining involves several essential steps, including data preprocessing, pattern discovery, and result interpretation. It plays a vital role in deriving insights from large volumes of data, which is particularly relevant in the context of Big Data. The relationship between Big Data and data mining lies in the vast amounts of data generated in various formats, such as structured, semi-structured, and unstructured data. Data mining techniques enable organizations to sift through this data deluge, identify hidden patterns, and extract valuable knowledge for decision-making and strategic planning.

Understanding the fundamentals of data mining is crucial in today's data-driven world, where organizations seek to harness the power of data to gain a competitive edge, improve operational efficiency, and enhance customer experiences. By exploring the concepts and methodologies of data mining, researchers and practitioners can unlock the potential of Big Data and leverage it effectively to drive innovation and growth



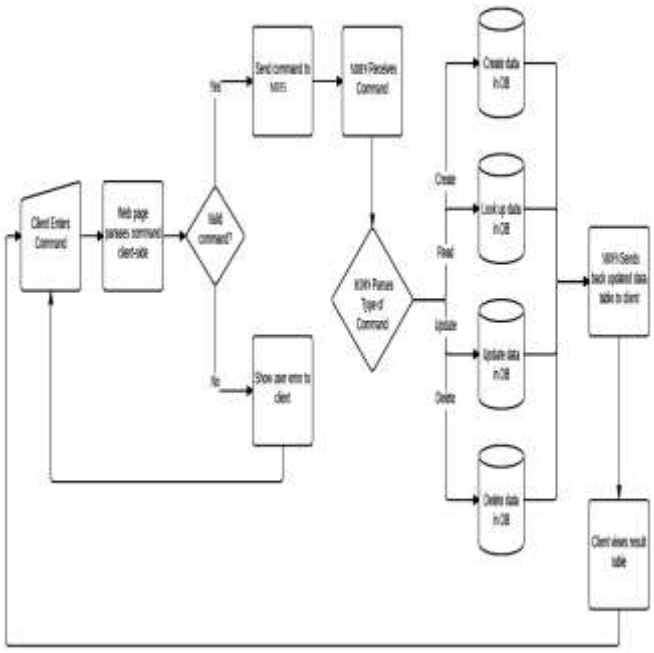
Pre-processing is a crucial step in data mining due to the inherent challenges posed by real datasets: they are often noisy, dirty, incomplete, and presented in various formats. Chapter 2 will delve into pre-processing techniques in detail, addressing methods to clean and standardize data for effective analysis.

The model refers to the techniques and algorithms utilized in data mining to extract insights from the data. A plethora of algorithms are available for this purpose, as outlined in [13], with common examples including K-means clustering and the A-priori algorithm. These algorithms serve as the backbone for uncovering patterns and trends within the dataset.

Validation marks the final phase of the data mining process, where the output patterns generated by the algorithms are scrutinized for accuracy and reliability. Not all patterns discovered are necessarily valid, hence the need for rigorous testing. This entails subjecting the data mining algorithms to a test dataset to ascertain if the output aligns with the expected results. If discrepancies arise, it prompts a re-evaluation of both the pre-processing and algorithmic steps to refine the analysis.
MDFS Sequence Diagram

In the MDFS (Massively Distributed File System) architecture, network bandwidth estimation between two nodes is facilitated by HDFS (Hadoop Distributed File System) based on their distance. The distance calculation involves considering the distance from a node to its parent node as one unit. By summing the distances to their closest common ancestor, the distance between two nodes can be determined. A shorter distance indicates a greater bandwidth available for data transfer between the nodes.
maximum of two replicas per rack where feasible. This distribution strategy optimally disperses block replicas across the cluster, mitigating risks associated with rack failures.

Once target nodes are selected, they are arranged into a pipeline based on their proximity to the first replica. Data transmission occurs in this order, optimizing the efficiency of data transfer operations. For reading operations, the NameNode verifies whether the client's host is within the cluster. If affirmative, block locations are relayed to the client in order of proximity to the reader. Subsequently, the block is read from DataNodes following this preference order, enhancing read performance and minimizing latency.



Functional Diagram

HDFS estimates the network bandwidth between two nodes by their distance. The distance from a node to its parent node is assumed to be one. A distance between two nodes can be calculated by summing the distances to their closest common ancestor. A shorter distance between two nodes means greater bandwidth they can use to transfer data.
MDFS allows an administrator to configure a script that returns a node's rack identification given a node's address. The NamesNode is the central place that resolves the rack location of each DataNode. When a DataNode registers with the NameNode, the NameNode runs the configured script to decide which rack the node belongs to. If no such a script is configured, the NameNode assumes that all the nodes belong to a default single rack.
The placement of replicas is critical to MDFS data reliability and read/write performance. A good replica placement policy should improve data reliability, availability, and network bandwidth utilization. Currently MDFS provides a configurable block placement policy interface so that the users and researchers can experiment and test alternate policies that are optimal for their applications.

The default MDFS block placement policy provides a tradeoff between minimizing the write cost, and maximizing data reliability, availability and aggregate read bandwidth. When a new block is created, MDFS places the first replica on the node where the writer is located. The second and the third replicas are placed on two different nodes in a different rack. The rest are placed on random nodes with restrictions that no more than one replica is placed at any one node and no more than two replicas are placed in the same rack, if possible. The choice to place the second and third replicas on a different rack better distributes the block replicas for a single file across the cluster. If the first two replicas were placed on the same rack, for any file, two-thirds of its block replicas would be on the same rack. After all target nodes are selected, nodes are organized as a pipeline in the order of their proximity to the first replica. Data are pushed to nodes in this order. For reading, the NameNode first checks if the client's host is located in the cluster. If yes, block locations are returned to the client in the order of its closeness to the reader. The block is read from DataNodes in this preference order.

This policy reduces the inter-rack and inter-node write traffic and generally improves write performance. Because the chance of a rack failure is far less than that of a node failure, this policy does not impact data reliability and availability guarantees. In the usual case of three replicas, it can reduce the aggregate network bandwidth used when reading data since a block is placed in only two unique racks rather than three.

FUNCTIONAL DIAGRAM

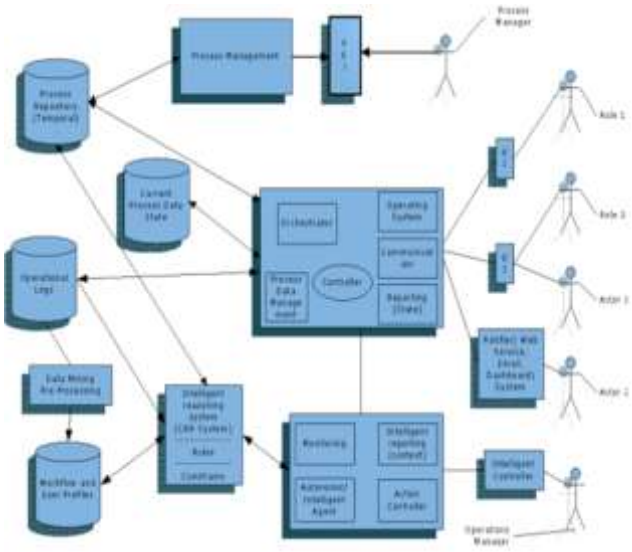


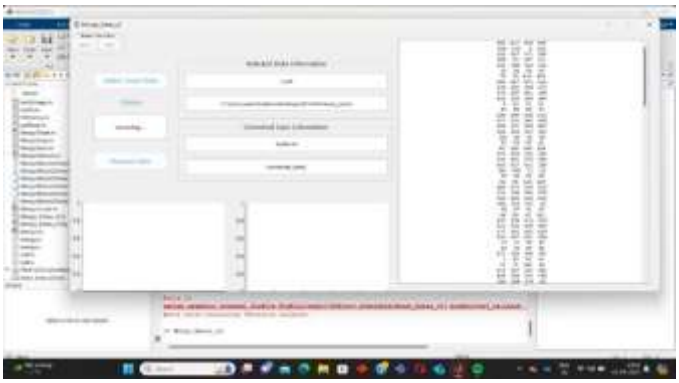
Fig.2 MDFS Architecture

MATLAB introduces a distributed file system along with a framework designed for the analysis and manipulation of extensive datasets, drawing inspiration from the MongoDB paradigm. While MDFS's interface resembles that of a traditional file system, adherence to standards was sacrificed to prioritize performance enhancements tailored to specific applications.

A defining feature of MATLAB is its ability to partition both data and computational tasks across numerous hosts, often numbering in the thousands, allowing for the parallel execution of application computations in close proximity to their respective data sets. MATLAB clusters can seamlessly scale computation, storage, and I/O bandwidth simply by adding commodity servers. These clusters, powered by MATLAB servers, can accommodate up to 40 Petabytes of application data, with the largest known cluster comprising 4000 servers. Furthermore, over a hundred organizations worldwide have reported leveraging MATLAB for their data analysis needs.

In MDFS, file system metadata and application data are stored separately. Following the model of other distributed file systems like PVFS, Lustre2, and GFS, MDFS designates a dedicated server known as the Name Node for metadata storage, while application data reside on other servers termed Data Nodes. All servers within the MDFS architecture maintain full connectivity and communicate via TCP-based protocols. Unlike Lustre and PVFS, MDFS's Data Nodes do not rely on mechanisms such as RAID for data protection. Instead, akin to GFS, file content is replicated across multiple Data Nodes to ensure reliability. This approach not only guarantees data durability but also enhances data transfer bandwidth while creating more opportunities for executing computations in close proximity to the required data setsModules

IV. MODULES



The system is divided into three main modules:

Data Layer:
The data layer serves as the interface for all data sources, which can include databases and data warehouse systems. Data mining results are stored in the data layer, allowing them to be presented to end-users through reports or visualizations.

Data Mining Application Layer:
This layer is responsible for retrieving data from the database. It may also include transformation routines to convert data into the desired format before processing it using various data mining algorithms.

Front-End Layer:
The front-end layer provides an intuitive and user-friendly interface for end-users to interact with the data mining system. Data mining results are presented in visualization form within this layer.

System Study

During the study of the system, several limitations were identified within MATLAB:

In MATLAB, when declaring item set values in an attribute portion, only those specific items are utilized in creating the data format. Failure to declare these similar item sets results in MATLAB displaying an error pop-up message, as it does not support undeclared numerical or string values.

In MATLAB, an associate class cannot be generated using lift or other metrics without confidence.

MATLAB does not provide results in a sorted order format, meaning that it generates all rules above the minimum support and minimum confidence level in a sequential manner from L(2) to L(n). This makes it challenging to separate the largest or most meaningful rules precisely, necessitating the manual checking of all possible largest rules from a vast number of generated rules.

MATLAB does not generate some types of interestingness measurements results (e.g., Certainty Factor, Relative Risk, Cosine, Information Gain, Ø-Coefficient, etc.) alongside specific rules. To obtain these measurements, manual calculation using appropriate formulas is required. Research Study

Research Aim: The aim of this study is to conduct a sample analysis of data management using data mining techniques suitable for processing the data.

Significance of Research:

A database management system (DBMS) is responsible for storing data and providing facilities for managing it. Modern database technologies, starting from relational databases and advancing further, are built on the principle of separating the logical representation of data from its physical instantiation. This allows for changes in one aspect without affecting the other. However, in many cases, the physical storage model aligns closely with the logical representation.

One of the key features of database management is the support for declarative access to data. This means that programs can specify what data they need without dictating how that data should be accessed. SQL (Structured Query Language) is commonly used for this purpose, although other declarative languages exist. To ensure optimal performance when accessing data via SQL, many DBMS incorporate an optimizer. The optimizer is capable of rewriting poorly written or generated code and determining the most efficient way to execute any given query.

VII.Results and Analysis:

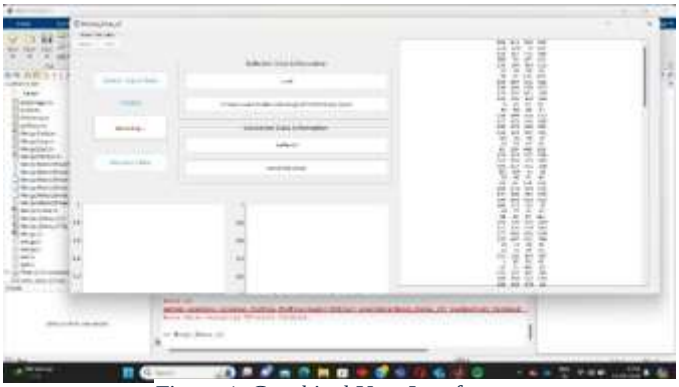
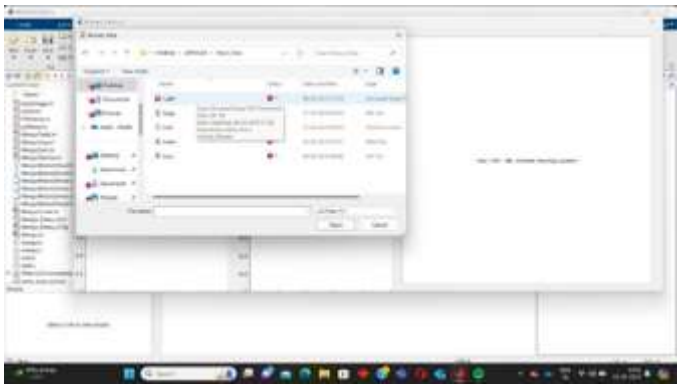


Figure 1. Graphical User Interface



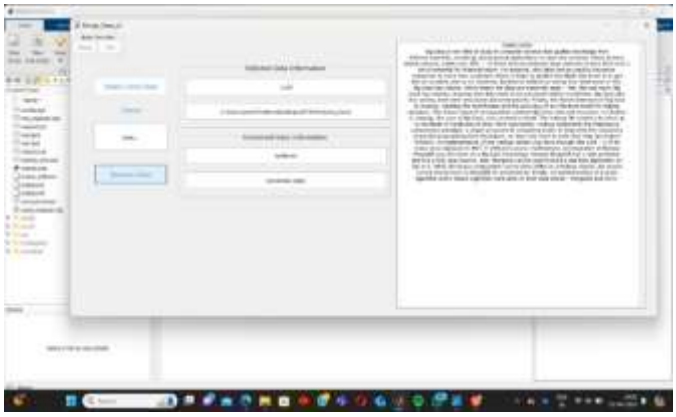


Figure 3. Processing Stage

VIII.CONCLUSION

In conclusion, Big Data is new field of study in computer science that applies knowledge from different scientific, technical, and practical applications to seek new answers. Smart phones, digital cameras, smart cars, GPS -- of these devices generate huge amounts of data that have a lot of potential for financial return. For instance, GPS data can be used by insurance companies to track their customers where it helps to predict how likely this driver is to get into an accident and so on.

Figure 4. Input data in MangoDB.

However, Big Data is defined as having four dimensions or 4Vs. Big Data has volume, which means the data are massively large – TBs, PBs and more. Big Data has velocity, meaning that data need to be processed almost in real time. Big Data also has variety, both semi-structured and unstructured.

This thesis research encapsulates common Big Data tools and concepts. CSP, the core of Big Data, was covered in detail. The CSP file system can store up to hundreds of Terabytes of data. More importantly, CSP implements the MapReduce computation paradigm, a simple yet powerful computing model. It helps hide the complexity of parallel programming from developers. An implementation of the CSP cluster was done through this work – 5 of its nodes were deployed on MET IT VMware's server.

Furthermore, an integration of CSP MongoDB was also done as a Big Data technology. CSP MongoDB has a high potential, and it is a fully open source. Also, MongoDB can be used to build a real time application on top of it. While the heavy computation can be done offline in a CSP cluster, the results can be stored back to MongoDB for presentation. Finally, an implementation of A-priori Algorithm and K-means Algorithm were done on both data stores – MongoDB and MFDS.

REFERENCES

[1] ErdiÖlmezoğulları, Ismail Ari, Online Association Rule Mining over Fast Data, 2013 IEEE International Congress on Big Data.

[2] Suthaharan, Shan, "Big data classification: problems and challenges in network intrusion prediction with machine learning." ACM SIGMETRICS Performance Evaluation Review 41.4 (2014): 70-73.

[3] Evans, Michael R., "Enabling Spatial Big Data via CyberGIS: Challenges and Opportunities." CyberGIS: Fostering a New Wave of Geospatial Innovation and Discovery, Springer Book, 2013.

[4] Mark J. Embrechts, "bigDAARE: Big Data Analytics for Renewable Energy", CFES 2012-2013 Annual Conference January 25, 2013.

[5] Anoop Verma, Andrew Kusiak, "Prediction of Status Patterns of Wind Turbines: A Data-Mining Approach", Journal of JSEE, JSEE (Journal of Solar Energy Engineering), February 2011.

[6] Kuncheva, Ludmila I., and Juan J. Rodríguez. "An experimental study on rotation forest ensembles." Multiple Classifier Systems. Springer Berlin Heidelberg, 2007. 459-468.

[7] Kale Suvarna Vilas, "Big Data Mining", Journal of CSMR, CSMR (International Journal of Computer Science and Management Research eETECME), October 2013.

[8] Mrs. Deepali KishorJadhav, "The New Challenges in Data Mining", Journal of IJIRCST, IJIRCST (International Journal of Innovative Research in Computer Science & Technology), September 2013.

[9] Rong Liu, Qicheng Li, Feng Li, Lijun Mei, Juhnyoung Lee, Big Data Architecture for IT Incident Management, 2014 IEEE.

[10] Han, Jiawei, Micheline Kamber, and Jian Pei, " Data mining: concepts and techniques: concepts and techniques." Elsevier, 2011.

[11] Minaei-Bidgoli, Behrouz, and William F. Punch. "Using genetic algorithms for data mining optimization in an educational web-based system." Genetic and Evolutionary Computation—GECCO 2003. Springer Berlin Heidelberg, 2003.

[12] Slimani, Thabet. "Application of rough set theory in data mining." arXiv preprint arXiv: 1311.4121 (2013).

[13] Zdzisław Pawlak, Roughsets And Data Mining, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, ul. Baltycka 5, 44 100 Gliwice, Poland.

[14] Hegland, Markus. "Data mining techniques." Acta Numerica 2001 10 (2001): 313-355.

[15] Mohammed J. Zaki, Limsoon Wong, Data Mining Techniques, August 9, 2003 WSPC/Lecture Notes.

[16] Freitas, Alex A, "A survey of evolutionary algorithms for data mining and knowledge discovery." Advances in evolutionary computing. Springer Berlin Heidelberg, 2003. 819-845.

[17] Ozer, Patrick, "Data Mining Algorithms for Classification." Radboud University Nijmegen, January 2008.

[18] Berkhin, Pavel, "A survey of clustering data mining techniques." Grouping multidimensional data. Springer Berlin Heidelberg, 2006. 25-71.

[19] Aloisioa, G., "Scientific big data analytics challenges at large scale" Proceedings of Big Data and Extreme-scale Computing (BDEC) (2013).

[20] Ularu, Elena Geanina, "Perspectives on Big Data and Big Data Analytics", Journal of DBSJ, DBSJ (Database Systems Journal) pp.3-14.

[21] Labrinidis, Alexandros, and H. V. Jagadish. "Challenges and opportunities with big data." Proceedings of the VLDB Endowment 5.12 (2012): 2032-2033.