WWJMRD 2023; 9(01): 1-6
www.wwjmrd.com
International Journal
Peer Reviewed Journal
Refereed Journal
Indexed Journal
Impact Factor SJIF 2017: 5.182 2018: 5.51, (ISI) 2020-2021: 1.361
E-ISSN: 2454-6615

Sowmya R
Research Scholar, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, VTU, Belagavi, Karnataka, India.

Dr. Suneetha K R
Professor, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, VTU, Belagavi, Karnataka, India.

Correspondence:
Sowmya R
Research Scholar, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, VTU, Belagavi, Karnataka, India.

Data Acquisition with Big Data using Rule-based Data Integration method

Sowmya R, Dr. Suneetha K R

Abstract
Data is an Information, in the form of facts or statistics that one can analyze or use for any further calculations in the respective domain to satisfy their own requirements. “Big Data” is a high-volume, high-velocity and high-variety information asset that requires new forms of processing for enhanced decision making, knowledge discovery and process needs. Big Data can be classified based on its characteristics that are used as per the designing data architecture for processing and analytics. Processing complex applications with large datasets. i.e., from terabytes to petabytes data in short time with a minimum cost is more challenging and problematic. Yet because this data is typically loosely structured and often incomplete, essential data is inaccessible to users. The proposed model mainly focuses on the insight view of the design model for the Data Acquisition stage in Big Data, which helps to achieve a perfect balance between providing data access control and facilitating data processing. The framework presents a systematic model to deliver Data Acquisition with Big Data using external data sources, events, and logs. The acquisition model in this presents the rule-based decision tree method to integrate the data from different sources.

Keywords: Big Data, Pre-processing, Decision trees, Data reduction, Cleansing, Data Integration, real-time.

1. Introduction

The rise in technology has led to the production and storage of voluminous amounts of data. Earlier megabytes were used but nowadays petabytes are used for processing, analysis, discovering new facts and generating new knowledge. Conventional systems for storage, processing, and analysis pose challenges in a large growth in the volume of data, various forms, and formats, increasing complexity, faster generation of data and the need for quickly processing, analysing and usage. Big Data is mainly defined by its 5*V*s fundamental characteristics. The 5*V*s include Velocity (data are growing and changing rapidly), Variety (data come in different and multiple formats) Volume (a huge amount of data is generated every second), Veracity (data quality) and Value (usefulness of data). Research communities from different sectors have been struggling to develop a new, fast, dynamic, and user-friendly technology for Big Data. Traditional searching methods are not adapted to the distributed environment and Big Data complexity. Enterprises need to run extensive real-time queries through huge volumes of unstructured and structured data sets. This demand has led to the development of scalable search engines based on appropriate searching and indexing technologies. Another important thing is to obtain the right data with the help of transformations in the Big Data. Given potential opportunities, Big Data presents ground-breaking discoveries through data-driven science and technology. Big Data may well be the next frontier for innovation, competition, and productivity. Since Big Data modelling and mining require powerful technologies and advanced methods to ensure performance, results in reliability, and scalability.

There are challenges even in Big Data, including how to capture, transfer, store, clean, analyse, filter, search, share, secure and visualize data. Even, storing and retrieving big data is one of the main problems in this area. Big Data problems require making several trade-offs among desired scalability, availability, performance, and security. Big Data starts with large volume, heterogeneous, autonomous sources with distributed and decentralized control

and seeks to explore the complex and evolving relationship among data. These characteristics make it an extreme challenge for discovering useful knowledge from Big Data.

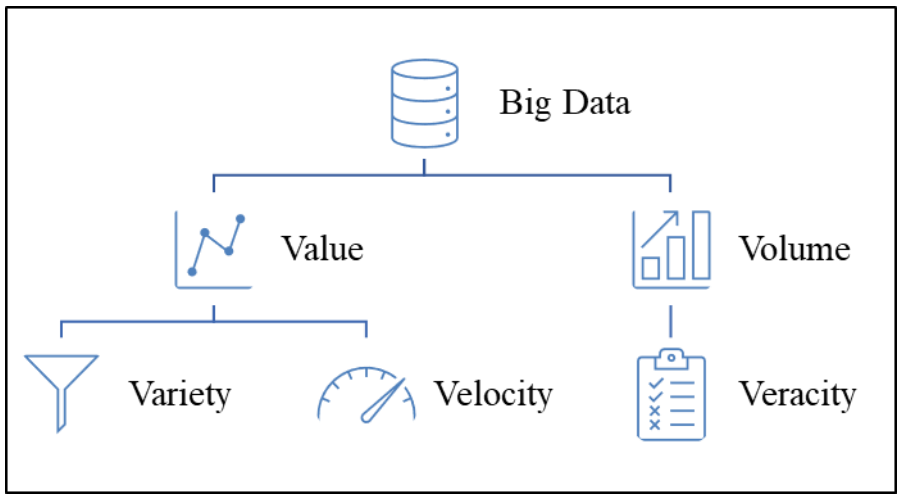


Fig 1: Big Data Characteristics

Volume: Size defines the amount or quantity of data which is generated from an application (s). The size determines the processing considerations needed for handling data.
Velocity: It refers to the speed of the generation of data. Velocity is a measure of how fast the data generates and processes. To meet the demands and challenges of Big Data, Velocity plays a major role.
Variety: Data is generated from multiple sources in a system. This introduces a variety in data and therefore which in turn introduces a 'complexity'. Data consists of multiple forms and formats. The variety is due to many heterogeneous platforms in the industry. This characteristic helps in the effective use of data according to their formats, thus maintaining the importance of Big Data,
Veracity: It refers to the quality of data captured. Quality is an important measure for data, and this can vary greatly, affecting its accurate analysis.
Value: It defines the usefulness of data and it's the result of veracity. The higher the data quality the more the value of data it holds.
The massive growth in the data is a key factor of the Big Data scenario. Addressing Big Data is a challenging and time-demanding task, which requires computational infrastructure to achieve successful processing and analysis. There has been a major growth and availability of data, which can be structured or unstructured. Most of the data is inaccessible to users, as we need technology and tools to find, transform, analyze, and visualize the data to make it for decision making. With the rapid growth of the internet of things, processing real-time data has become a great challenge for research and applications.
This paper is organized as follows: Section 2 provides the main methods for the data acquisition framework. Section 3 provides details on the data pre-processing stage since the data pre-processing is a major part of the data acquisition. Section 4 contains the conclusion and future work details.

2. Methodology

The objective of this paper is to provide a framework for the Data Acquisition process in Big Data. Data Acquisition refers to the process of gathering information and is subdivided into 3 stages i.e., data collection, data transmission and data pre-processing. In the first stage, Data collection refers to the process of obtaining actual data in real-time using dedicated data collection technology [1]. In the second stage, collected data should be transmitted into the specific storage system for various analytical or real-time systems for further processing. This requires a high-speed data transmission mechanism. Collected data sets from the first stage might contain meaningless or irrelevant data, which impacts the amount of storage space and gives inaccurate data analysis in further steps. In some of the data collected, the most common issue is Redundancy. Thus, in the third stage efficient storage and mining are performed to retain the relevant data as pre-processing operations. In the world of Big Data [3], more value is for time. As time goes by, the value of knowledge contained in the data is very important, especially in real-time systems. Real-time Big Data provides an opportunity for researchers to explore and more on this. The main challenge in the acquisition of data is Variety [5]. Variety has existed as a popular character in Big Data. Since some of the data pre-processing systems are not suitable for the high-performance needs of real-time applications. In real-time applications, the processing is happening on the fly due to the dynamic nature of data. The Data Pre-processing stage is a major role, this stage is even split into substages. This paper represents a Pre-Processing stage in detail. The proposed framework [Fig 2] in this paper consists of the following stages:

1. Data Collection.
2. Data Transmission.
3. Data Pre-processing.

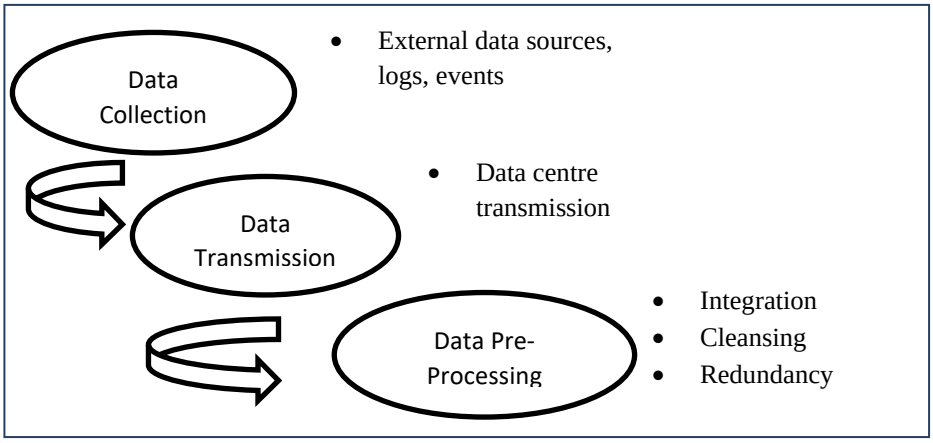


Fig 2: Data Acquisition Framework

3. Workflow of the framework

1. Data Collection

Data collection refers to the process of retrieving data from real-world objects. The process needs to be well designed. Otherwise, inaccurate data collection would impact the data analysis procedure which leads to invalid results. Currently, different data collection methods are available. The methods considered in this proposed framework are External data sources such as APIs and log files, even this can be extended for other methods further for any other applications.

Data collection is through the distributed agent/proactively collected by a centralized process. So, it is also known as Pull based approach.

The most deployed data collection method in almost all the applications is Log files or via external APIs. A data source system generates the files to record the transactions or activities in a pre-defined file format for further analysis. All the applications running on digital devices are using Log Files. For example, an access log file in a web server records the hits, clicks, attributes and access details at a user level. The network monitoring data, stock ticks in financial applications, and traffic management applications are all using the Log file-based data collections method. In this method, data are pushed from its source third party to the respective sink. So, it is also known as Push based approach. One more collection method via API [Application Programming Interface], is software in between two applications to talk to each other. Most of the time in this context data collection from an external path is to pull the data from an API by building a URL. It's known as Pull based approach. The data collection framework is flexible to handle both push and pull-based approaches.

2. Data Transmission.

In this stage, actual data which has been collected from the previous stage is transferred into specific defined data storage. The data centers contain different data storage infrastructures. The transmission procedure is described as follows.

1. A high-capacity internet scale is required to transfer the data from its source to a data centre. The IP backbone infrastructure helps to achieve the data transmission. But the capacity and rate of transmission are depending on the respective link management protocols and the physical media.
2. Big Data will be transited to the data centre after transmission for further processing. This method is known as a data centre transmission.

time restrictions. Even though UDP is used for large data transfer but lacks congestion control. There are even different categories of API protocols such as REST, RPC, and SOAP architectures available. Based on the application framework, the protocol selection needs to be taken.

3. Data Pre-Processing.

Since the collected data sets are from different diverse sources. The data quality levels vary in different measures such as redundancy, noise, and consistency. The reason for main challenge for the Big Data pre-processing procedure is due to the dynamic features of application data, which are huge, heavy, and multi-source. There is much research has been carried out on Big Data pre-processing, but the research about real-time Big Data pre-processing has not been given much importance. In real-time, the incoming data rate is always huge, and the data keeps coming every second. Due to this, there will not be much time to do a separate pre-processing task. The pre-processing tasks should be performed at the initial extract-transform-loading with a minimal amount of time. Otherwise, delay in pre-processing impacts the results delivery. Thus, there is a need to work on technology to transform and visualize the data sets and tools to analyze the consumable data which helps to solve the decision-making with more accuracy. Big Data engineering is crucial and a major challenge and scope for the research communities. In the Big Data acquisition framework, data integration is of a heterogeneous world and is a difficult task. The major resource for the researchers is to work with schema-less and complex data sets integration methods. Data Pre-processing is further divided into 3 stages 1. Integration 2. Cleansing and 3. Data Reduction.

4. Data Integration using the Rule based Decision trees Method.

The data integration execution procedure is to combine data present in different sources and provide users with a uniform view of data. There are some approaches to data integration, in that two approaches can be considered as important in this paper. One of the approaches is the data warehouse logical procedure, also known as Extraction, Transformation and Loading [ETL] and the data federation method. ETL consists of 3 steps [4] i.e., Extraction, Transformation and Loading. The data federation procedure is also called data virtualization in which a virtual database is created from a disparate source. The

query or aggregation can be applied to the virtual database. The virtual database contains the details of the metadata about the actual data with its position/place, not the data itself. The two methods such as ETL and data federation use the "store and pull" approach, which is not powerful for the search or streaming applications to achieve the required performance. Since the streaming data is real, dynamic and queries are processed on the fly. Srividya K. Bansal et. al [2] demonstrated a Semantic Extract-Transform-Load framework, which generates semantically linked data as Resource Framework Triples (RDF) that meets the standards of specific the data model based on some defined integrated datasets. This framework has been tested in a couple of case studies. But the framework works based on domain-specific ontology, alignment, and mapping of data fields. This framework has not been tested for real-time data processing. One more challenge is, to requires creating multiple ontologies with alignment and mapping of data fields for multi-source data sets. So, this model is not suitable for real-time pre-processing in Big Data.

Our paper proposes a Rule Based-Decision tree to the ETL framework [Fig 3], which generates a defined uniform model of the data sets required to be integrated and then generates data that compiles to that data model. A custom-based rules decision tree is applied at the transformation step, which helps to classify and integrate the data from the different sources. In the data extraction step, required data attributes from the different sources are extracted and then applied the custom data validation rules in different functions for different data sets. A decision tree is a model of the tree-like structure, in which each node presents a data quality validation test on an attribute value. Each branch denotes an outcome/result of the data validation test, and tree leaves represent a data model or group of data models. Data quality classification rules can be built using the Decision trees [6]. Generally, in real-time data pre-processing it's not required to load and store the initial/actual data always. Since the focus will be on the processed data and its prediction more than on the initial captured data. So, the rules at the extraction step help to generate only the transformed data for further processing.

Decision tree classification is commonly used in operations research, specifically in decision analysis. This helps to compute the conditional probabilities. One of the main advantages of this approach is, that it can be easily coupled with other decision techniques. For the data acquisition with a more unstructured pattern, Decision trees create a comprehensive analysis of the consequences along each branch and identify decision nodes that need further analysis. In this context of Data Acquisition in Big Data, the new customs rules can be added to the Decision trees easily without any impact on the existing structure.

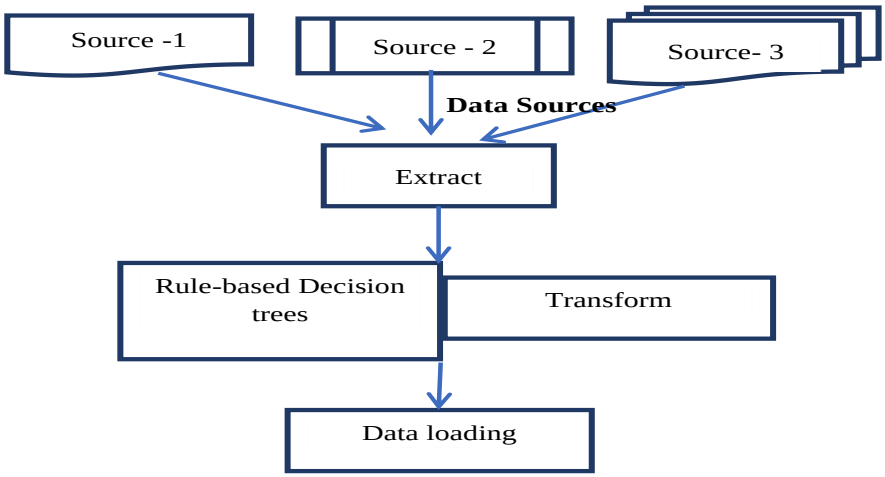


Fig 3: Data Integration Framework.

Fig 4 represents the construction of a sample decision tree using the above-mentioned attributes classification and decision rules from a twitter output API. Twitter output API is unstructured and contains a lot of information on every trigger.

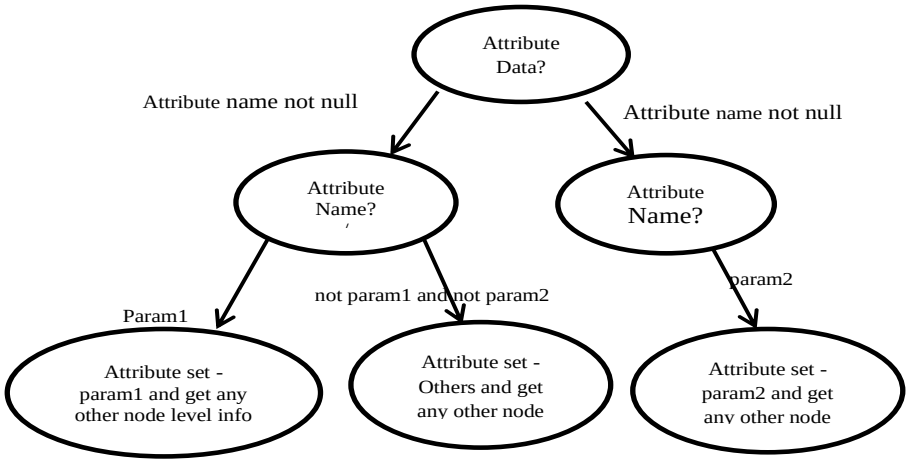


Fig 4: Decision tree construction.

Rule 1: If Attribute Value is not null and Attribute name is B and Attribute data value contains the string as “param1”. Then the Data value needs to be considered as a param1 attribute.
Rule 2: If the Attribute Data value is not null and Attribute name is D and the Attribute value contains the string "param2”. Then the Data value needs to be considered as a param2 attribute.
This model helps to provide the flexibility that the customized rules can be applied based on the application domain and grouping using the decision trees and the templates, which in turn helps to select only the focused set of attributes based on the data model. Thus, instead of consuming the unwanted data, this method helps to pre-process the required set of attributes.

4.1. Cleansing

The rules applied during the data transformation help to determine the inaccurate data, but there are chances of unreasonable data in real-time data ingestion and pre-processing. To make sure the data quality, a separate cleansing step is required for the loaded data after the extraction transformation and loading. Cleansing of data [7] helps to improve the analysis accuracy. In this framework, cleansing checks consist of the below steps:

1. Define and document the error types based on the attribute list.
2. A separate check for the error instances.
3. Define the standard procedure to handle the missing values based on the attribute list and the data model.

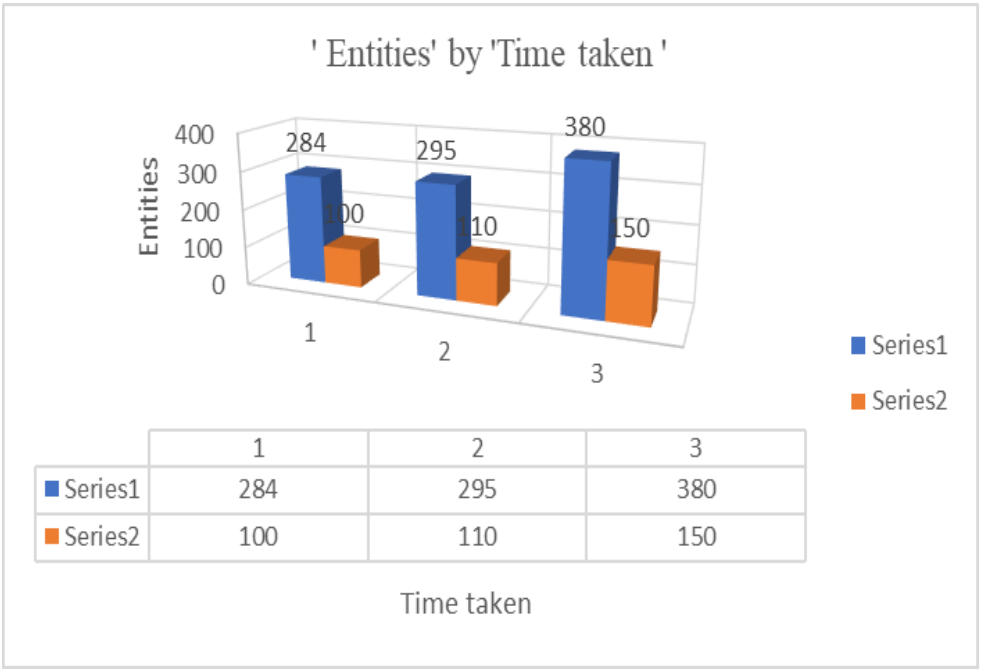
4.2. Redundancy Elimination

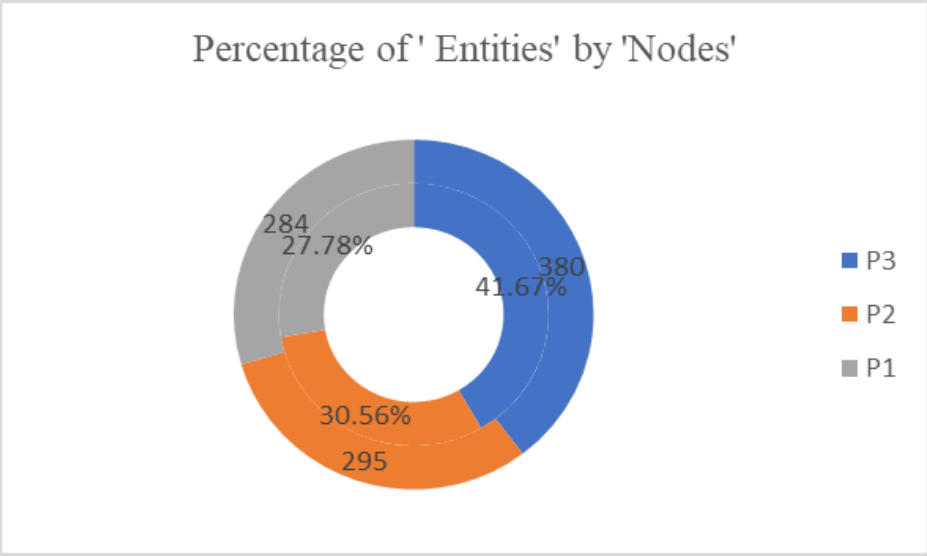
Data redundancy is the recurrence of data in the same data model, which is caused by the data integration step. The real-time processing of data offers a new and greater requirement for the timeliness of data [8]. Real-time data processing can happen in batch, stream, and combination of two mixed processing modes based on the application. To support the real-time mode of processing, this framework proposes an attribute subset selection in which redundant, irrelevant, and invalid attributes are detected and removed. There are situations in some applications, where we can store or load the captured data. During these cases, the redundancy elimination method helps to find a minimum set of attributes such that the resulting data model of the same classes is as close as possible to the original data sets obtained using all the attributes. The Attribute construction helps to understand the data quality and improves the structure of the data model.

4. Results

Internal validation and testing: The rule-based decision framework has been validated using the data extracted from Twitter API. The finance related data using the index symbol has been extracted into csv files from the Twitter API in real time. The transformation logic using the rule-based decision trees helps to store the data individually at the index symbol level for further processing. The key for the decision tree is the index symbol which has been used for the transformation and the extraction purposes as per the above-mentioned workflow. The results obtained from this framework are provided in the below table. The framework validates against the different nodes with the variant resources and compares the results obtained.
P1: Memory: 8 GB, CPU Base Speed = i5 - 2.70 GHz
P2: Memory: 16 GB, CPU Base Speed = i7 - 2.95 G
P3: Memory: 8 GB, CPU Base Speed = i5 - 2.70 GHz

Twitter API			
Nodes	Data Entities	Output files	Run-time (Sec)
P1	100	100	284
P2	110	110	295
P3	150	150	380





Conclusion

With the focus on dynamic and real-time data processing, the proposed Data Acquisition model has been designed to work with Big Data. Currently, the above framework is implemented using Python. Many platforms are available to implement the proposed model in Big Data Technologies. The same framework can be enhanced to work with other Big Data Technologies such as Apache Spark, an advanced version of Hadoop, is built using in-memory primitives for faster performance in distributed computing. This has the mechanism of loading data directly into memory and keeps reusing based on the requirements. Spark is based on the framework called Resilient Distributed Datasets (RDDs), a new abstraction model. This archaic nature of this model helps in managing the data partition, maintaining persistence and many more features. As a future work for this paper, this can be implemented for the proposed Data Acquisition model using the Apache Spark technology and evaluate the results in real time using even multiple sources.

Note: No Funding has been granted.

References

1. Han Hu, Yonggang Wen, Tat-Seng Chua and Xuelong, “Toward Scalable Systems for Big Data Analytics: A Technology Tutorial”, IEEE Access, Vol.2, July 2014.
2. Srividya K Bansal, Sebastian Kagemann, “Integrating Big Data: A Semantic Extract-Transform-Load Framework, IEEE Computer Society”, Vol.48, Issue 3, March 2015.
3. Xindong Wu, Xingquan Zhu, Gong-Qing Wu, Wei Ding, “Data Mining with Big Data”, IEEE transactions on Knowledge and Data Engineering, Vol.26, No.1, January 2014.
4. Jaydeep Chakraborty, Aparna Padki, Srividya K.Bansal, “Semantic ETL-State of the art and open research challenges”, IEEE 11th International Conference on Semantic Computing, 2017.
5. David Ostrowski, Nestor Rychtyckyj, Perry MacNeille, Mira Kim, “Integration of Big Data Using Semantic Web Technologies", IEEE 10th International Conference on Semantic Computing, 2016.
6. K.R.Suneetha, R.Krishnamoorthi, “Classification of Web log data to Identify Interested Users Using Decision Trees”, International Conference on Computing, Communications and Information Technology Applications, (CCITA 2010), Coimbatore, India.
7. Salvador Garcia, Sergio Ramirez-Gallego, Julian Luengo, Jose Manuel Benitez and Francisco Herrerra, “Big Data preprocessing: methods and prospects”, Vol 1, Page 1, Big Data Analytics, 2016.
8. Sowmya R and Suneetha K R, "Data Mining with Big Data," 2017 11th International Conference on Intelligent Systems and Control (ISCO), 2017, pp. 246-250, DOI: 10.1109/ISCO.2017.7855990.