Data modeling techniques used for big data in enterprise networks

Article *in* International Journal of Advanced Technology and Engineering Exploration · April 2020

DOI: 10.19101/IJATEE.2020.762029

2 authors:

Richard Omollo
Jaramogi Oginga Odinga University of Science and Technology
8 PUBLICATIONS 7 CITATIONS
SEE PROFILE

Sabina Alago
Jooust
2 PUBLICATIONS 4 CITATIONS
SEE PROFILE

Review Article

International Journal of Advanced Technology and Engineering Exploration, Vol 7(65)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2020.762029

Data modeling techniques used for big data in enterprise networks

Richard Omollo[*] and Sabina Alago
Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Bondo, Kenya

Received: 10-March-2020; Revised: 18-April-2020; Accepted: 20-April-2020

Abstract

The deployment and maintenance of enterprise networks form the bedrock of any organization, be it government, commercial, academic, and/or non-profit making. These networks host vast amounts of information, databases, in either temporary mode while in transit or permanent mode while stationary. The databases are managed by the information systems with appropriate functions that meet consumers' needs. Databases hold varying data – structured, semi-structured, or unstructured. Data is increasingly becoming a vital organizational asset and therefore plays a crucial role in making organizational decisions. With growth in the internet, digital data sources have become ubiquitous. In turn, this has seen the continued growth in the volume, variety, veracity, velocity, and value of data. Big data brings with its data complexities that have an eventual impact on the data modeling techniques. This paper presents a review of big data modeling techniques with a concentration of enterprise networks. We started by appreciating big data before embarking on modeling techniques for big data.

Keywords

Data, Model, Modeling techniques, Big data, Enterprise networks, Databases.

1.Introduction

Data is continuously being used to make decisions that are transforming institutions [1]. Today, institutions are increasingly harnessing the potentials brought about the rapid advancements in technology. One of such potentials includes the rapid data growth both historical and current [2]. Currently, many institutions are apt in making efficient use of the tremendous data in realizing strategic organizational objectives including: improved business, improved organizational output and productivity among many objectives. Therefore, this paper sets forth underpinning issues around big data modeling [3].

Institutions are sensitive in terms of institutional data, and this is clearly exhibited by their preparedness to turn challenges into opportunities. Data has become an important driver for revolutionizing institutions, thus, a corporate asset.

In as much as this is the case, it is also important to note that, the data are quickly becoming complex in terms of volume, velocity, variability, veracity, valence, value, variety, integrity, availability, reliability, confidentiality among other characteristics [1, 3]. These complexities have an eventual contribution to data modeling techniques and technologies in terms of enabled capture, storage, distribution, management and analysis of data [4─6].

Each organization is peculiar in terms of its data needs. These data needs vary from one institution to the other. And as such, database needs also vary. The database can be defined as a collection of sharable data that originates from any kind of digital source [7]. The value of data as an organizational asset cannot be downplayed. The ability to understand and manage the value of the data must be in place. Lack of this means data becoming a liability rather than an asset. This absurdity is in itself a vehicle that drives the need for an optimally functioning database management system. Database management systems have become ubiquitous and are fundamental tools in the management of information. Therefore, data modeling is an integral part of database management systems [1, 8].

[*]Author for correspondence

Databases hold varying data – structured, semi-structured, or unstructured. Structured data highly adheres to a pre-defined data model and such data are easy to aggregate, store, process and access locally and remotely. Unlike structured data, unstructured data have no pre-defined model and are highly characterized by lots of ambiguities, irregularities, making them difficult to understand using traditional structured data models, programs and databases. Semistructured data do not conform to either structured or unstructured data models, but rather rely heavily on self-describing data structure - separates semantic elements and enforces hierarchies of data [7]. It is therefore imperative to model data before applying analytics to any tremendous amounts of data. Data modeling plays an important role in big data analytics [1].

The data model can be defined as a high-level data description that hides low-level description storage details. Database management systems allow user-defined data to be stored in terms of the data model. The data model can be viewed as concepts that describe the structure of a database – structure of how data is held. Database management systems can be based on a relational, non-relational, network, hierarchical data models amongst others [1], [8]. Notably, data modeling comes in handy when there is a need to design data structures at various levels of abstraction [1]. Additionally, the authors suggest that effective database designs (logical, conceptual and physical) are grounded on effective data modelling efforts.

a) Objectives of the study

In many application domains, the triviality of the data modelling techniques cannot be overlooked. Data modeling techniques allow structures that are suitable for representing data. This paper sought to put forth the working definitions of big data. Agreeably, large volumes of data are generated continuously via diverse digital data sources and in order to obtain newer insights, knowledge of big data definitions, technologies and analytics is imperative. With the quickly soaring data, organizations need to be informed on the role of big data in achieving their predefined functions. Data can only be regarded as an asset only if its value to the organization can be ascertained. This partly has explained the adoption of diverse functional definitions of big data in varied contemporary settings.

The second concern that the paper addresses is big data structures and big data modelling techniques. Data need to be well structured. Inferring data structures behind big data storage can be somewhat misleading, knowing big data, data structures beforehand goes a long way in facilitating our understanding of big data management. Different enterprises, deal with varying sets of data. Does this always mean they are dealing with complex models?
Lastly, the paper brings forth the natural relationship between bid data and modelling techniques. Data can be voluminous but what is their relationship between data models and data volumes? It is important to have a grounded knowledge on such inferences if they are to be made. For instance, it is possible sensor data can be very voluminous but with simple data structures compared to graph structures that are quite complex.

In summary, this paper forms a modelling point of view on big data and proposes conformity of data to the models developed for an enterprise network.

b) Limitations of the study

In present day, the request for big data demands for non-relational database. Notably, we have seen that the choice of correct database is strongly linked to storage demands by any enterprise network. These are points to note for any enterprise network during its operations. However, in this paper, we have noticed the difference between different models, but not, the difference in the models in terms of records held within the databases manifesting the different models. Indisputably, databases are widely used in enterprise networks and as such, handling of the same should be of paramount interest for any organization. If this is so, then modelling of the data is definitely tagged along. Some of the things to consider while modelling include amount of data, flexibility of the model, budget, amount of transactions, and frequency of transactions.

However, this being the case, we acknowledge that this study does not show the results of different operations that have been applied by particular databases – relational and non-relational. This is in terms of execution of the operations- manipulations that are essential in simultaneous user interactions in an enterprise network.

c) Methodological approach used in the study

Our key focus was based on big data that is resident on enterprise networks, which appropriate data modeling techniques aids in analyzing. We acknowledged that there are various data modeling perspectives, methods, and techniques that can be easily misconstrued by researchers and scholars in data science. The contribution of this paper was to review these key concepts with a view of adopting a systematic approach to gather and critically evaluate the concepts around data modeling on big data that is the main component of an enterprise network. We started by critically analyzing various dimensions of big data used in its definitions before we embarked on reviewing several data models and applicable data modeling techniques. Given that our research design integrates a review of existing concepts and follows an interpretivism research philosophy, we appreciated the significance of modeling big data that resides in enterprise networks. We acknowledge that proper understanding data modeling techniques adopted are beneficial to decision making in an organization.

2.Discussions on big data and its data modeling techniques

Although there are many technical papers on big data modelling techniques, there is still need to continually describe big data, data modelling, and modelling techniques used for big data. Remember, the current global agenda largely revolves around big data. With this in mind, despite relatively little being written on their deployment and / or application in enterprise networks, it does not mean we should continue folding our hands in this regard, but rather continuous exploratory works on the same is called for if the gap in big data modelling techniques is to be bridged.

That being said, during this study, we observed several exceptions that are worth being mentioned: [1] provides a deeper understanding of big data modelling techniques and points out some developments that have been made in this regard. The author's contributions include the role that big data modelling has in the management of data sources. Another study, [2], highlights the value of big data analytics in reforming enterprise networks. For instance, in this case, there's a clear demonstration on how big data are reforming higher education, particularly the learning activities and processes. A related study by Ale [4] contributes to this global agenda by noting that continuous development of such techniques results into operational big data models. The paper emphasizes on the need of continuous big data analytics, newer and enhanced technological innovations if the world is to keep abreast with contemporary societal issues. Lastly, a related study [7], recommends that like universities, other enterprise networks need to adopt newer data technologies and data models, harness their potential so as to meet the customer needs that comes along with proper management of big data.

In summary, newer and / or enhance big data technologies, including data models are called for. For this reason, we sought to contribute to this ongoing global discourse.

a) Big data and its characteristics

The use of the term "big data" was officially adopted in the computing field in the year 2005 when Roger Magoulus from O'Reilly tried to describe the huge amounts of data that could not be managed and processed by the traditional data management techniques because it was becoming too complex and vast in size [9, 10]. Even though Roger has been credited for coining the term "big data", some sources also claim that it had been used before by John Mashey of Silicon Graphics in the 1990s. The originator of the term may not be the focus of this paper, but the root description of the term "big data" that helps in establishing a more functional understanding is. In our review, we found more clarity in Roger's description since it facilitates clearer understanding of his definition and thus it enabled us to view "big data" in the context of its characteristics.

Despite Roger's viewpoint, the term "big data' can still be easily associated with voluminous data, that is, quantified data that measure in zettabytes or even yottabytes. Many do not comprehend the fact that there are other definitions, which are beyond stored contents in computing devices. This point of understanding is majorly facilitated by large data sets that stream in storage devices within computing networks. According to current estimates by World Economic Forum, the amount of data in the world stands at 44 zettabytes in the year 2020. Undoubtedly, this means that in the coming years, there is a possibility of having huge amounts of data that will reside in the computing devices and networks. However, comprehensive definition of big data can be viewed in tandem with the incorporation of other attributes of data as summarized in the sub-sections below.

i. **Big Data in the Three Dimensions, 3 Vs:** In reference to [11, 12], the three Vs (volume, velocity, and variety) are considered as the core focus when defining big data and this is in synchrony with Madden's [13] definition of big data that says: it is data that is too big (volume) coming from different sources, too fast (velocity) as it must be processed quickly, and too hard (variety) to be processed by existing tools. Doug Laney viewed big data in three dimensions and his description has been widely accepted by data, scientists in the key data management challenges [12, 14].

ii. **Big Data in the Four Dimensions, 4 Vs:** There is another approach in defining big data by expanding on the 3 Vs to 4 Vs where the aspect of *veracity* (too uncertain data) is incorporated from the 3 Vs that exists in the earlier description [15]. The four Vs was proposed by IBM scientists while trying to explain traffic patterns and downloads of data that end up being recorded, stored and analyzed to enable technology and services relied upon daily. The position fronted here is that depending on the type of industry and organizational enterprise networks, big data encompass information derived from both internal and external sources. Organizations thus opt to leverage data to adapt their products and services to satisfy their customers' needs, optimize their operations and infrastructural resources, and enhance their new sources of revenue.

iii. **Big Data in the Five Dimensions, 5 Vs:** It has been argued that there is no precise definition of big data especially when its characteristics are considered [16]. From the *Figure* 2 below, another attribute, *value*, is introduced to give it an enhanced meaning, thus the five Vs of big data. Significantly, big data are important to organizations because it aids in gathering, storage, management and manipulation of vast amounts of data for making useful decisions. This qualifies the inclusion of "value" as the fifth attributes of big data from the added-value of collecting data enforces intended process or predictive analysis. The data value is related closely to other attributes of big data like volume and variety [17].

iv. **Big Data in the Six Dimensions, 6 Vs:** In an enterprise network, big data analytics accommodate many perspectives: business, technology, and social. These perspectives comprise both functional and non-functional requirements and therefore, big data impacts both the strategy development process and the actual strategies developed in a number of ways [18]. In the previous definitions of big data, the attributes of big data remain the same in an enterprise network apart from *variability* that take consideration of the network infrastructure especially on integration, changing data and model [19].

v. **Big Data in the Seven Dimensions, 7 Vs:** We have seen from the previous definition of big data that by introducing more attributes, a refined understanding is realized by applying on an environment. Remember, we are now focusing on big data that is non-relational, and which traditional data management techniques and tools cannot handle. From the last definition, we have seen the introduction of variability, which takes care of varying data and associated models. More understanding of non-relational data sets like in spatial and temporal database models that deal with data representation calls for a refined description. This incorporates the seventh attribute, which is *a visualization* [20]. This attribute accommodates readability and accessibility of data presentations which require numerous spatial and temporal parameters, and associated relationships between them [21].

vi. **Big Data in the n^{th} Dimensions, *n* Vs:** As it has been demonstrated in the previous sub-sections, the definition of big data can continue to accommodate more attributes till the n^{th}, where n is the number of attributes incorporated. This is still a prime research field for scholars with the goal of ending up with a more refined definition. We may not put a caveat on this, but encourage more research endeavours to explore better positions on big data definition. Nowadays, there are many V's considered in attempts to define big data in a more refined way [22]. For instance, the *eight Vs of big data* introduces *viscosity* that addresses the concern of *data sticking and/or calls for actions.* In the case of *the nine Vs of big data* covers *volume (size of data), velocity (speed of generating data), variety (different types of data), veracity (accuracy of data), vocabulary (data models and semantics), venue (distributed heterogeneous data), variables (dynamic data), validity (quality of data), value (useful data)* and *vagueness (confusions over the meaning of data).* Other studies have discussed the *ten Vs of big data* that see the incorporation of *volatility* that is concerned with the life duration of data – for how long will data be considered valid, and for how long should it be stored [23]? This description of data in terms of the characteristic has also attracted divergent views and approaches. For instance, in [24], the researchers have introduced the components of *big data intelligence* that is about the set of ideas, technologies, systems and tools that can mimic human intelligence in the management and processing of big data. This has been considered due to its strategic role in improving competitiveness of business performance because of the support it offers in decision making e.g. the cases of Google, Facebook [25] etc. The *big data analytics* have also been considered as another key characteristic of big data since it encompasses the process of collecting, organizing and analyzing big data to discover and visualize patterns, knowledge and intelligence for decision making [26]. Like in the case of apache Hadoop ecosystem, *big data infrastructure* has also been argued to be an important characteristic that forms the *ten Vs of big data* since it captures all structures, technologies, systems, platforms, and facilities that serves big data processes [24].

We appreciate the fact that many researchers have different views in approaching the understanding of big data definitions, for instance, the 42 Vs of big data characteristics [27]. We are also aware that there are contrary opinions on the relevance of Vs characteristics in regard to defining big data [28] and thus we do not want to tie one to this line of thinking. We acknowledge overwhelming arguments that support the integration of the Vs characteristics in realizing functional understanding and definitions of big data. Despite all these, variants of big data definitions, see *Figure 1*. Madden [13] definition still presents the most stable understanding of big data, that is, at the end of the day, it is still about how big, how fast and how hard the data is. We concur that more research work on a comprehensive definition of big data can still be pursued, especially in the context of its characteristics.

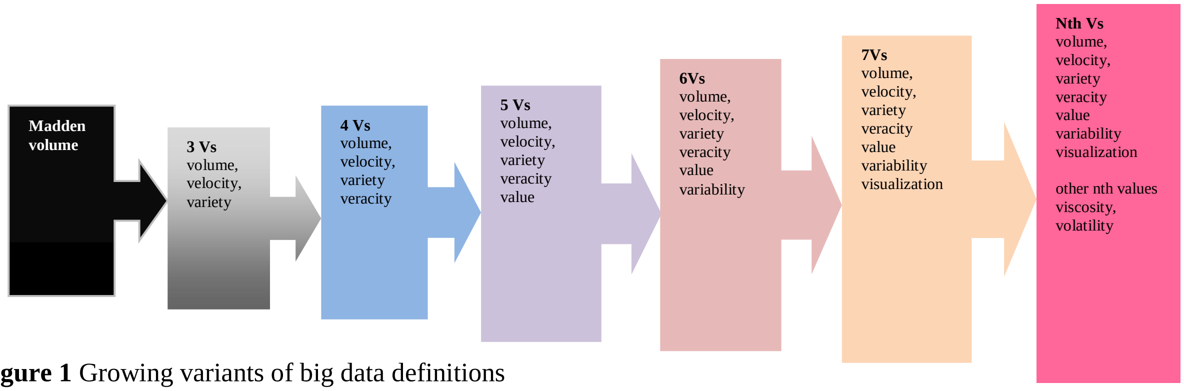

Figure 1 Growing variants of big data definitions

b) Data modeling techniques

As earlier discussed, indeed, big data increasingly becoming a corporate asset, thus the calls for continuously harnessing of the potentials that come with it [2]. There are multiple digital sources of big data, including the Internet, social networks, mobile devices, emails, blogs, videos, organizational transactions and other consumer interactions [29]. With this in mind, organizations, therefore need to put in place measures in terms of power, but functional tools to store and explore the tremendous amounts of data. Notably, users of such tremendous amounts of data have their expectations around availability, reliability, confidentiality, integrity, accuracy among other data security goals [4─6]. We still iterate data is not gathered for the sake of it. Its value to the organization must be determined upfront and through at usage time. This whole process is captured in a summarized form as illustrated in *Figure 2* below. First, identify the type of data - data collection phase; secondly, data preparation stage, where data is cleaned from ambiguities, anomalies, redundancies, inconsistencies and so on; thirdly, data modelling stage, where the evaluation of different modelling techniques is done; fourthly, and then the best model that suits for our work is selected. After modelling deployment is done. Thereafter, data is stored for immediate or later usage – meaning the value of the data to the organization has been properly determined. This fourth phase may be referred to as a data deployment phase. Lastly, the next step is a data analytics phase. Here, appropriate big data analytics tools are deployed to help determine data trends, make a prediction. They help to further harness the potentials that come with big data.

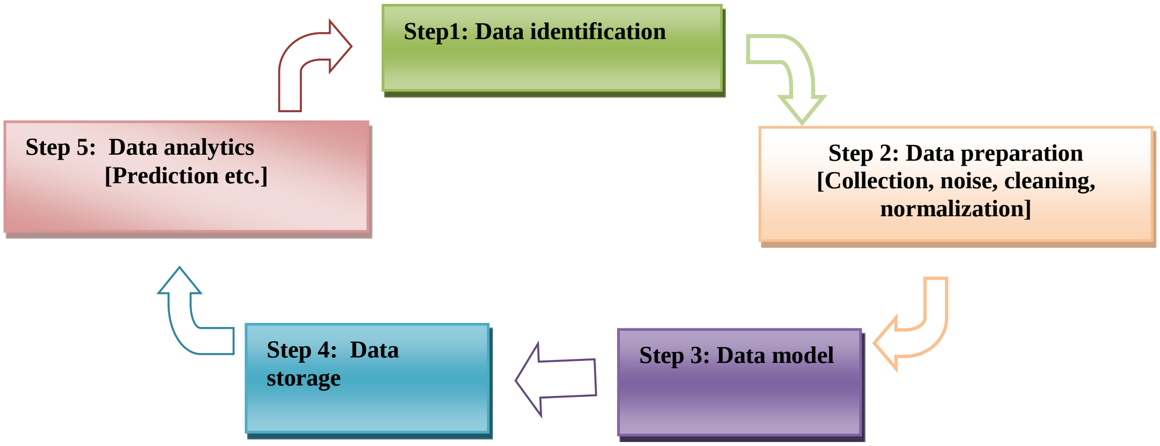
Step1: Data identification
Step 2: Data preparation [Collection, noise, cleaning, normalization]
Step 3: Data model
Step 4: Data storage
Step 5: Data analytics [Prediction etc.]

Figure 2 Big data modelling and analytics process

Data modeling can loosely be translated to mean creating a model for storage of data in a database, or more precisely a theoretical presentation of data objects and association among various data objects within a particular database. The end product of data modeling that supports business information infrastructure (enterprise network) is called data model, and the expertise behind this achievement is a data modeler. The data model can deeply be viewed as a set of conceptual tools for describing relevant properties of the system under consideration [30]. It consists of three distinct but closely intertwined components, namely;

- A notation for data structure description
- A set of constraints that data values must satisfy in order to be considered valid, and
- A set of operations for data update and retrieval.

Data model plays significant role in enterprise network as it bridges the gap between technical and functional areas in an organization and thus ensures all data objects required by the database are represented accurately. There are majorly three perspectives [31] of viewing data model, namely;

- **Conceptual data model**: the highest level of abstraction that defines *what* the system contains, and its main focus is to organize, scope and define business concepts and rules.
- **Logical data model**: the medium level of abstraction that defines *how* the system should be implemented devoid of any database management system (DBMS), and its main focus is to develop technical map for rules and data structures, and
- **Physical data model**: the lowest level of abstraction that describes how a system will be implemented using a specific database management system (DBMS), and mainly focuses on the actual implementation of the database.

Data modeling is important because it guarantees a clear representation of data, making it easier to analyze data that is properly represented in a model. In addition, this improves data quality that in turn enables concerned stakeholders make data-driven decisions that are useful for the organization. There are several ways of achieving data modeling, some of the methods [32] include:

- **Hierarchical data model**: this is a tree-like data modeling where each of the records has a single root or parent. The hierarchical order is used as a physical order for storing the database.
- **Network data model**: Built on the hierarchical model where it allows multiple relationships among records. This means that it has multiple parent records with multiple child records. Again, each record can belong to multiple sets allowing the model to be an effective one for conveying complex relationships.
- **Relational data model**: here, data segments is combined with the help of tables and detailed knowledge of physical data storage is adopted by an organization on its network. It was combined with Structured Query Language (SQL) as its de facto database language.
- **Object-Oriented (O-O) data model**: this consists of objects each having features and methods. It can be referred to as post-relational or hybrid database models since it is not limited to tables though it uses tables. Examples that suit this include multimedia, and hypertext databases.
- **Entity-Relationship data model**: this model graphically represents entities and relationships. It

can also be referred to as Entity-Relationship Diagram (ERD).

- **Object-Relational data model**: this is also called Extended Relational Data Model (ERDM) since it is a hybrid - simplicity of the relational data model with some of the advanced functionalities of the object-oriented database model. Examples of Database Management Systems (DBMS) include Object/Relational DBMS (O/RDBMS), and the language is SQL3 with interface call interfaces are ODBC, JDBC, and proprietary call interfaces.
- **Semi structured data model**: This is a self-describing data model where information usually associated with a schema is contained within the data [32]. By its flexible design, it contains typed scheme [33].
- **Inverted file data model**: this model is built with the inverted file structure so as to facilitate fast full text searches with data contents being indexed as a series of keys in a look up table.
- **Flat data model**: this is non-scientific data model that simply lists all the data in a single table. Access and manipulation is achieved when computer reads the entire flat file into its memory.
- **Record base data model**: This data model is used to specify the overall structure of the database and has many record types. Here, each record has a fixed number of fields with fixed length.
- **Multi- dimensional data model**: In this data model, structural data usually contained in the database schema is embedded with the data itself. It is useful for handling web-based data sources and also for describing interactions between databases that do not adhere to the same schema.
- **Context data model: T**his flexible data model since it incorporates other data models as needed - elements of relational, object oriented, semi-structured and network data model are cobbled together enabling different works to be done due to its versality. This model supports different types of users who differ by interactions in the database.
- **Associative data model:** in this data model, all the data points are divided based on whether they describe an entity (anything that exists independently) or an association (something that only exists in relation to something else). It structures data as two sets: set of items (a unique identifier, a name and a type), and set of links (each with unique identifier of sources, texts and target).
- **NoSQL data models:** these are non-SQL data models that emerged in contrast to the relational model. They include:

    - **Graph data model:** that allows any node to connect to any other. This is more flexible than network data model.
    - **Multivalued data model:** this breaks relational model by allowing attributes to contain list of data rather than a single data.
    - **Document data model:** This is for handling storage and management of documents or semi-structured data rather than atomic data.

The methods listed in the data models above are as a result of data modeling. These are different from modeling techniques since each method may attract a unique technique, for instance, there are three basic data modeling techniques that applies to relational and non-relational models, namely;

- **Entity Relationships Diagram (ERD)**: this visual technique is like the default for modeling and design relational databases. It incorporates the use of entities, attributes, relationships, cardinalities and participation constraints among other components as seen case by case. It involves adoption of different types of symbolic notations too e.g. Crow's Foot, Chen etc.
- **Unified Modeling Language (UML) class diagrams**: this is a standardized family of notations for modeling and designing information systems resident in enterprise networks [32]. Different aspects of the system are represented by different set of notations as it applies in software engineering, and class diagrams are like ERDs in real world and mostly applicable in designing classes in object-oriented programming languages e.g. Small talk, Java etc. Each Class Diagram incorporates classes, attributes, methods, relationships between objects and classes, among other components per each case. This technique is synonymous with implementation of meta-model, the *New Data Model (NDM)*, when migrating conceptual model to Object Relational Database (ORDB) [34].
- **Data dictionary**: this is a tabular representation of data sets, with its core elements being data sets and attributes [35]. They have detailed specifications and can complement well with ERDs. Other expanded considerations on its elements include items descriptions, relations between tables and additional constraints e.g. uniqueness, default values, value constraints or calculated columns [32]. The concept of *active data dictionary* is

when data model and application independent part of program code combine to serve as compatible program shell controlled by its data model. It is useful in the case of ERDM or ORDM [30].

From the data models presented above, we summarily grouped them according to convergence in their characteristics and suitable modeling techniques as in the *Table* 1. We appreciated that some data models display overlapping characteristics and may be grouped under more than one consideration. Our clustering was premised on the design principles exhibited in each data model.

Table 1 Clusters of data models

No.	Cluster	Data models	Modeling techniques
I	Record Based	Hierarchical, Network, Inverted File, Record based, Flat, Multidimensional,	ERD, Data Dictionary
II	Relational	Dimensional, E-R, Enhanced E-R, ERDM	ERD, Data Dictionary
III	Object Based	E-R, O-O, ERDM, Context,	UML, ERD
IV	NoSQL	Graph, Multi-value, Semi Structured, Associative	Data Dictionary

Day by day, traditional data modeling tools are facing their limits, especially, when it comes to handling of big data. As a result, newer and/or enhanced data modeling techniques are being adopted to help model data as they grow in volume, value, veracity and variety [29]. In big data analytics, there are two main categories of tools [36]:

- **Univariate big data analysis tools**: that engage single variable and useful in forecasting, growth rate and instability indices
- **Bivariate big data analysis tools**: that establish the relationships between two variables as in conjoint analysis, predictive modeling, and
- **Multivariate big data analysis tools**: that involve more than two variables and more pronounced in artificial neural network, cluster analysis, propensity score modeling among others.

Data modeling techniques and tools come in handy in capturing and translating complex systems designs into easily understood representations of the data flows and processes. They also go a long way in creation of database blue print for actual construction or re-engineering. It is therefore possible to deploy multiple modeling techniques in viewing similar sets of data to help ascertain all processes, entities, relationships and data flows have been identified and captured [29].

In the previous sections, we have observed that big data goes beyond structured data as exhibited by its characteristics. That is why we approached big data modeling with a map onto non-relational data modeling, where we discussed each family. Non-relational data modeling referred to also as NoSQL data modeling has not been well studied compared to relational data modeling. This has been exhibited by lack of systematic theories on the same. Databases, NoSQL Databases, derived from this modeling approach get identified by various number of non-functional criteria like their scalability, performance and consistency.

We considered approaching this non-relational data modeling from a systematic viewpoint of NoSQL data models thus showing the trends and interconnections. An imaginary "evolution" of the major non-SQL (No SQL or Not Only SQL) systems families captured by [37] consists of, namely; key-value stores, big table-style databases, document databases, full text search engines, and graph databases.

We note that the acronym NoSQL got its use in 1998 by Carlos Strozzi while naming his Strozzi NoSQL Open Source Relational Database, a light open source "relational" database that did not use SQL [38]. Later in 2009, Eric Evans and Johan Oskarsson came to use it for describing non-relational databases. The main motivation behind the development of NoSQL was to address web data that necessitated faster processing of unstructured data. The NoSQL data model uses a distributed database management system (DDBMS) thus is quicker, uses ad-hoc approach for organizing data, and process large amount of differing kinds of data. Again, NoSQL databases can handle large and unstructured data sets better than relational databases and this is attributed to their speed and flexibility. Therefore, NoSQL systems not only handle both structured and unstructured data but they also process unstructured big data quickly [39].

The non-relational databases, premised on NoSQL data model, are built on the Consistency, Availability, Partition (CAP) Theorem [40], which guides the simultaneous establishment of the three guarantees i.e. consistency, availability, and partition tolerance, and thus quantifies tradeoffs between Atomicity, Consistency, Isolation, Durability (ACID) and Basic Availability, Soft state, Eventual consistency (BASE) models. Consistency, for instance, is adopted by both ACID and BASE models [41] even though BASE model provides less assurance in providing safe environment for processing data compared to the ACID model. Most NoSQL databases use the ACID constraints for ensuring safe and consistent storage of data. On the other hand, availability of scaling purposes, which is an important feature of BASE data stores, enables it to be used by aggregate stores [39, 42]. Therefore, non-relational data storage, which is often open-source, non-relational, schema-less and horizontally scalable, uses BASE model for consistency. This elasticity feature allows for rapid changes and replication, something that has been accomplished by designing NoSQL data storage from bottom up and optimized for horizontal scaling [39]. It is evident that modeling with non-relational data systems is completely different from modeling used in relational data systems because of its reliance on different design philosophies. In conclusion, NoSQL systems uses data stores optimized for specific purposes and can be achieved in one of the four categories; key-value, document, wide-column, and graph database.

We started by briefly describing each NoSQL System family before exploring the characteristics, merits and demerits for each case. This enabled us to appreciate their suitability in adoption in different practical scenarios of big data cases, as summarized in *Table 3.*

i) key-value stores-based/database model

Description

This adopts data storage system designed for storage, retrieval, and managing associated arrays. Unlike relational databases, it prioritizes a variety of optimal options when clarifying data types.

Characteristics

- It is a non-relational model.
- It consists of a pair, a unique key that can be used to identify data and a value. This means that, a row contains several columns each with a pair of unique column value and column key which are used to identify unique data [1, 29]
- It is based on a big table that stores contents and comments from whence data can be queried

Benefits

- Largely used with social networks
- It is suitable for fast retrieval of information [43].
- It improves aggregation capabilities [29].
- It limits the requirement of formatted data for storage. Meaning data can be primitive, string, integer, an array or an object [3].
- There is no need for fixed data model [3].
- Can be partitioned across distributed servers

Challenges

- Focus is on the ability to store and retrieve data rather than the structure of that data [29].
- Poor applicability to cases that requires processing of key ranges [37].

ii) Document stores/database model

Description

This is a system designed for storage, retrieval, and managing document-oriented information. It has similarities to key-value store apart from the way the data gets processed i.e. it uses internal structure of document for identification and storage [39]. It also advances the BigTable model by improving on values with schemes of arbitrarily complexities, and database managed indexes in some implementations.

Characteristics

- They are schema-less and querying is based on multiple attribute value constraints [3].
- It extends the key-value model; key values are stored in a structured format that the database can understand.
- Originally intended to store traditional documents.
- Horizontal scalability and sharding across the cluster nodes [44]

Merits

- Supports more complex data compared to key/value based- supports secondary indexes and multiple types of objects
- Good for content-oriented applications (with a single query entire content can be fetched)
- Can store huge collections of textual documents
- Good also for semi structured data or de-normalized data [43].
- Enhanced to support any domain object [43].
- Easily find and reuse information through metadata [45].
- Quickly disseminate contents to many recipients.

Demerits

- Not suitable for ACID transactions [44]

iii) BigTable/column-oriented stores model

Description

Also referred to as column family databases, it focuses on groups of columns for storing, processing and accessing data.

Characteristics

- Data values are modeled as map-of-maps-of-maps, namely, column families, columns, and time-stamped versions [29].
- Column keys in Big Table get grouped together as column families and data within a family shares a similar data type [29].
- Within a specified column family, data is stored row-by-row, with columns for a specified row being stored together instead of each column being stored individually [37].
- Horizontal Scalability [44]

Merits

- Column oriented layout is very effective to store very sparse data as well as multi-value cell [29].
- Data within a column family can share the same data type [29].
- Makes reads and writes more efficient in a distributed environment because big tables are distributed in nature [29].
- Suitable for batch- oriented data processing [43].
- Suitable for large scale data processing [43].
- Suitable for exploratory and predictive analytics [43].
- Unlike in row-oriented database models, it has better capabilities to manage data and storage space [44].

Demerits

- Manage data as sections of columns rather than rows like in relational database management systems [43].

iv) Graph-oriented stores model

Description

It evolved from graph theory [46] which is designed to represent entities and their relationships as nodes and edges respectively [47]. Unlike in traditional databases or other NoSQL stores, graph databases embrace connected and semantically strong relationships between large amount of data [44]. It is considered as a side branch of evolution that origins from the ordered-key value models [37].

Characteristics

- Some graph database exploit native graph storage and native processing [44].
- Are schema-less non-relational databases [48]
- Data are represented as a network of nodes (domain entities) and are connected by edges (connections/relationships between the entities) [10].
- Properties are represented as key-pair values - a node is used as a unique identifier to and from each node there are edges and these form a pair of key values – together they are a pointer to a relationship [29].

Merits

- Designed to handle complex relationship data – relationships that are unpredictable and slow in nature.
- Handy when analyzing interconnections.
- Can be used in data mining
- Can be used in mapping a document
- Adopts visual paradigm -user friendly in nature because of their visual representation [43].
- Graph data can be queried more efficiently because intensive joins are not necessarily required in graph query languages [44].
- Allow one model business entity transparently related to document database because many implementations allow one model a value as a map or document. [37]

Demerits

- Are slow
- Are based on identified entities

v) Natural language processing (NLP) and ontology ecology or full text searches engines

Characteristics

- Ontology model is a classification of entities and models the relationship among those entities
- NLP is to identify the entities and understanding of relationship among those entities. [29].
- Models the unstructured data - Unstructured data is stores specific format – meets grammatical rules
- Based on grammar to express thoughts and extract information.
- Uses natural language processing text.

Merits

- Enables computers makes sense of human language
- Making sense of natural language enhances inferencing [29]

c) A comparative analysis of big data modeling techniques

We have successfully presented, in the previous sections, an understanding of big data and its characteristics. We acknowledged that the definitions of big data by various authors follow particular patterns of integrating dimensions to enrich the understanding of this fundamental concept. We noted that the definitions may vary but the core of its roots is premised on Madden [13]. This presentation enabled us to view modeling of big data in a simpler manner, and thus we effectively outlined the various modeling techniques that applies for unstructured data. It is clear for structured data, which calls for modeling approaches synonymous with relational databases, but becomes complex with data that are catered for by non-relational (NoSQL) databases. In the *Table 2* below, we managed to classify various data modeling techniques that are used in big data cases.

After studying various data modeling techniques use in big data for enterprise networks, we concluded based some parameters. There have been favoured considerations while modeling data and these apply to big data case, namely;

- It is advisable to consider traditional techniques to see if they can handle the big data in question
- The modeler is also required to design system or schema that apply for the situation
- Consideration of big data modeling tools thus choosing an appropriate one to use
- The data modeler should also focus on data that is core to organization's business and its philosophy
- Not to deviate from the real business of data modeling where the main goal is quality of data
- The data modeling techniques applied should appreciate performance, cost, efficiency and quality

In our summary analysis of data modeling techniques, we came up with a guideline that can be adopted when viewing appropriate data modeling techniques in big data for an enterprise network. We integrated the parameters (performance, cost, efficiency, quality) in good modeling techniques applied in the various big data cases by ranking their suitability under *High (H)*, *Medium (M)* and *Low (L)*. The summary is in *Table 3* below.

Our analysis in *Table 3* above interpreted the performance of the data modeling, the cost (resource wise) of implementing the techniques, how efficient is the technique and does it provide quality modeling results. This was our proposal based on the understanding of the considered parameters of data modeling techniques. We agree that there is need for more research work on this with empirical approach in implementation of various data modeling on different data sets in different computing environment. We know that different big data sets require special considerations and some even calls for combination of approaches for better results.

Table 2 A comparative analysis of NoSQL data modeling techniques

Modeling techniques	Specific modeling	Descriptions	NoSQL Model
Conceptual (Basic Principles)	Denormalization	Query processing simplification / optimization	Key-value, Document, BigTable
	Aggregates	Soft schema, complex internal structures	Key-value, Document, BigTable
	Application Side Joins	Design time joins vs query time joins	Key-value, Document, BigTable, Graph
General (General techniques for NoSQL implementation)	Atomic Aggregates	Guaranteed atomicity, locks, test-&-set instructions	Key-value, Document, BigTable
	Enumerable Keys	Generated sequential IDs, daily buckets	Key-value stores
	Dimensionality Reduction	Multi-Dimensional data mapping	Key-value, Document, BigTable
	Index Table	Special tables with keys	BigTable Style
	Composite Key Index	Multi-dimensional indexing	BigTable Style
	Aggregation with Composite Keys	Records aggregation indexing	Ordered key-value, BigTable
	Inverted Search – Direct Aggregation	Data aggregation, criteria indexing	Key-value, Document, BigTable
Hierarchy (index based and mapping)	Tree Aggregation	Modeling arbitrary graphs	Key-value, Document
	Adjacency Lists	Independent Node Modeling	Key-value, Document
	Materialized Paths	Denormalization, node	Key-value, Document,Search Engines

Modeling techniques	**Specific modeling**	**Descriptions**	**NoSQL Model**
		attribution to identifiers	
	Nested Sets	Leafs Storage, Leafs Mapping	Key-value, Document
	Nested Documents Flattening: Number Field Names	Business entities to plain documents mapping	Search engines
	Nested Documents Flattening: Proximity Queries	Proximity queries for acceptable distance limitations	Search Engines
	Batch Graph Processing	MapReduce, Message Passing Pattern	Key-value, Document, BigTable

Table 3 A comparative ranking of NoSQL data modeling techniques

Modeling Techniques	**Specific Modeling**	**Performance**	**Cost**	**Efficiency**	**Quality**
Conceptual (Basic Principles)	Denormalization	M	L	H	H
	Aggregates	M	H	M	H
	Application Side Joins	M	M	M	H
General (General techniques for NoSQL implementation)	Atomic Aggregates	M	L	H	M
	Enumerable Keys	M	M	M	H
	Dimensionality Reduction	M	H	M	H
	Index Table	H	M	H	H
	Composite Key Index	H	H	H	H
	Aggregation with Composite Keys	H	M	M	H
	Inverted Search – Direct Aggregation	H	M	M	H
Hierarchy (index based and mapping)	Tree Aggregation	H	L	H	H
	Adjacency Lists	M	M	H	H
	Materialized Paths	M	L	H	H
	Nested Sets	H	M	H	H
	Nested Documents Flattening: Number Field Names	M	M	M	M
	Nested Documents Flattening: Proximity Queries	M	M	M	M
	Batch Graph Processing	M	L	M	M

3.Conclusion and future work

Indeed, new and/or enhanced data management techniques are imperative if big data potentials are to be optimally harnessed. New knowledge is needed for working around big data and enterprise network systems. Where enterprise network systems are, interdisciplinary teams are inevitable. And as such,

- Understanding big data on enterprise network systems perspective is key - data structure and data models.
- Besides, any new or enhanced horizon of big data knowledge is needed, especially with an inclination towards enterprise networks and computational modeling when tackling voluminous data - user interactions with the database.

In this paper, we have set apart the big data modeling techniques in enterprise network systems. Data characteristics do not necessarily infer complexities in the data model. We also introduced another angle of interpreting various data modeling techniques as applicable for big data by ranking them against the parameters and attributes.

We also have observed that it is important to identify the logical dependencies between data entities, that is how data is classified, organized and stored. Appropriate data modeling techniques offer an appropriate storage environment, thus allowing optimal performance while meeting data security goals. It would be of utmost importance if more studies on Intelligence on data models are done. Suppose they will be helpful in the management of data, including those in enterprise network systems.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Jyothi BS, Jyothi S. A study on big data modelling techniques. International Journal of Computer Networking, Wireless and Mobile Communications. 2015; 5(6):19-26.

[2] Tulasi B. Significance of big data and analytics in higher education. International Journal of Computer Applications. 2013; 68(14):21-3.

[3] M. M. Huda, M. L. Hayun and Z. Martun. Data modelling for big data. ULTIMA Infosys. 2015; 6(1):1-11.

[4] Ale B. Risk analysis and big data. In Safety and Reliability 2016 (pp. 153-65). Taylor & Francis.

[5] Baumann SU, Erber IR, Gattringer MA. Selection of risk identification instruments. ACRN Oxford Journal of Finance and Risk Perspectives. 2016; 5(2):27-41.

[6] Chen J, Tao Y, Wang H, Chen T. Big data based fraud risk management at Alibaba. The Journal of Finance and Data Science. 2015; 1(1):1-10.

[7] Logica B, Magdalena R. Using big data in the academic environment. Procedia Economics and Finance. 2015; 33:277-86.

[8] Patel A, Patel J. Data modeling techniques for data warehouse. International Journal of Multidisciplinary Research. 2012; 2(2):240-6.

[9] Chaorasiya V, Shrivastava A. A survey on big data: techniques and technologies. International Journal of Research and Development in Applied Science and Engineering. 2015; 8(1):1-4.

[10] Ularu EG, Puican FC, Apostu A, Velicanu M. Perspectives on big data and big data analytics. Database Systems Journal. 2012; 3(4):3-14.

[11] Zhu J, Wang A. Data modeling for big data. CA, Beijing. 2012.

[12] Austin C, Kusumoto F. The application of Big Data in medicine: current implications and future directions. Journal of Interventional Cardiac Electrophysiology. 2016; 47(1):51-9.

[13] Madden S. From databases to big data. IEEE Internet Computing. 2012; 16(3):4-6.

[14] Laney D. 3D data management: controlling data volume, velocity and variety. META Group Research Note. 2001.

[15] Saabith AS, Sundararajan E, Bakar AA. Parallel implementation of Apriori algorithms on the hadoop-mapreduce platform-an evaluation of literature. Journal of Theoretical and Applied Information Technology. 2016; 85(3):321.

[16] Hadi HJ, Shnain AH, Hadishaheed S, Ahmad AH. Big data and five v's characteristics. In IRF international conference 2014.

[17] Anuradha J. A brief introduction on Big Data 5Vs characteristics and Hadoop technology. Procedia Computer Science. 2015; 48:319-24.

[18] Demchenko Y, De Laat C, Membrey P. Defining architecture components of the Big Data Ecosystem. In international conference on collaboration technologies and systems 2014 (pp. 104-12). IEEE.

[19] Lněnička M, Máchová R, Komárková J, Čermáková I. Components of big data analytics for strategic management of enterprise architecture. In SMSIS: proceedings of the 12th international conference on strategic management and its support by information systems 2017. Vysoká škola báňská-Technická univerzita Ostrava.

[20] Alexandru A, Alexandru C, Coardos D, Tudora E. Healthcare, big data and cloud computing. Management. 2016; 1:2.

[21] Malik BH, Cheema SN, Iqbal I, Mahmood Y, Ali M, Mudasser A. From cloud computing to fog computing (C2F): the key technology provides services in health care big data. In international conference on material engineering and advanced manufacturing technology 2018 (p. 03010). EDP Sciences.

[22] Patgiri R, Ahmed A. Big data: the v's of the game changer paradigm. In international conference on high performance computing and communications; IEEE international conference on smart city; IEEE international conference on data science and systems 2016 (pp. 17-24). IEEE.

[23] Khan N, Alsaqer M, Shah H, Badsha G, Abbasi AA, Salehian S. The 10 Vs, issues and challenges of big data. In proceedings of the international conference on big data and education 2018 (pp. 52-6).

[24] Sun Z, Strang K, Li R. Big data with ten big characteristics. In proceedings of the international conference on big data research 2018 (pp. 56-61).

[25] Sun Z, Wang P, Strang K. A mathematical theory of big data. IEEE Transactions on Knowledge and Data Engineering. 2017; 13(2):83-99.

[26] Sun Z, Sun L, Strang K. Big data analytics services for enhancing business intelligence. Journal of Computer Information Systems. 2018; 58(2):162-9.

[27] Farooqi MM, Shah MA, Wahid A, Akhunzada A, Khan F, ul Amin N, Ali I. Big data in healthcare: a survey. In applications of intelligent technologies in healthcare 2019 (pp. 143-52). Springer, Cham.

[28] https://tombreur.wordpress.com/2018/12/16/the-three-vs-of-big-data-or-four-five-seven-10-or-42/. Accessed 15 December 2019.

[29] Hashem H, Ranc D. An integrative modeling of bigdata processing. International Journal of Computer Science and Applications. 2015; 12(1):1-15.

[30] Mišić V, Velašević D, Lazarević B. Formal specification of a data dictionary for an extended ER data model. The Computer Journal. 1992; 35(6):611-22.

[31] https://www.guru99.com/data-modelling-conceptual-logical.html. Accessed 15 December 2019.
[32] Liu L. Encyclopedia of database systems. Özsu MT. New York, NY, USA: Springer; 2009.
[33] Chakraborty S, Chaki N. A survey on the semi-structured data models. In computer information systems–analysis and technologies 2011 (pp. 257-66). Springer, Berlin, Heidelberg.
[34] El Alami A, Bahaj M. The migration of a conceptual object model com (conceptual data model CDM, unified modeling language UML class diagram...) to the object relational database ORDB. MAGNT Research Report. 2(4):318-2.
[35] Ermolayev VA, Keberle NG. Active data dictionary: a method and a tool for data model driven information system design.2000.
[36] Ramadas, Sendhil. (2017). Big data analytics: tools and approaches. ICAR-Indian Institute of Wheat and Barley Research. 2017; 1-4.
[37] https://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/. Accessed 20 December 2019.
[38] https://www.datastax.com/blog/2018/08/evolution-nosql. Accessed 20 December 2019.
[39] https://www.dataversity.net/a-brief-history-of-non-relational-databases/. Accessed 20 December 2019.
[40] https://dzone.com/articles/understanding-the-cap-theorem. Accessed 20 December 2019.
[41] https://www.dataversity.net/choose-right-nosql-database-application/. Accessed 20 December 2019.
[42] Pritchett D. Base: an acid alternative. Queue. 2008; 6(3):48-55.
[43] Ribeiro A, Silva A, Da Silva AR. Data modeling and data analytics: a survey from a big data perspective. Journal of Software Engineering and Applications. 2015; 8(12):617.
[44] Farooq H, Mahmood A, Ferzund J. Do NoSQL databases cope with current data challenges. International Journal of Computer Science and Information Security. 2017; 15(4):139-46.
[45] https://www.dataversity.net/graph-database-vs-document-database-different-levels-of-abstraction/. Accessed 20 December 2019.
[46] Besta M, Peter E, Gerstenberger R, Fischer M, Podstawski M, Barthels C, Alonso G, Hoefler T. Demystifying graph databases: analysis and taxonomy of data organization, system designs, and graph queries. arXiv preprint arXiv:1910.09017. 2019.
[47] Ramachandran S. Graph Database Theory Comparing Graph and Relational Data Models. https://www.lambdazen.com/assets/pdf/GraphDatabaseTheory.pdf. Accessed 20 December 2019.
[48] Srinivasa S. Data, storage and index models for graph databases. In graph data management: techniques and applications 2012 (pp. 47-70). IGI Global.

Dr. Richard Omollo, PhD, is a faculty member and researcher in the Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Kenya. His research interests include Programming Systems, Modeling, Information Security, IP Networks and Artificial Intelligence.
Email: comolor@hotmail.com

Sabina Alago, MA, is an adjunct faculty and researcher in the Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Kenya. Currently, she is working on a research study in Big Data where she endeavors to utilize Bayesian Methods for Big Data Security in Enterprise Networks.