Received February 13, 2018, accepted March 15, 2018, date of publication April 2, 2018, date of current version May 24, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2820176

A Methodology of Real-Time Data Fusion for Localized Big Data Analytics

SOHAIL JABBAR[1], KALEEM R. MALIK[2], MUDASSAR AHMAD[1], OMAR ALDABBAS[3], MUHAMMAD ASIF[1], SHEHZAD KHALID[4], KIJUN HAN[5], AND SYED HASSAN AHMED[6], (Member, IEEE)

[1]Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan
[2]Department of Computer Science and Engineering, Air University Multan Campus, Multan 66000, Pakistan
[3]Faculty of Engineering, AlBalqa' Applied University, Amman 19117, Jordan
[4]Department of Computer Engineering, Bahria University, Islamabad 44220, Pakistan
[5]Department of Computer Engineering, Kyungpook National University, Daegu 41566, South Korea
[6]Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, USA

Corresponding author: Kijun Han (kjhan@knu.ac.kr)

This work was supported in part by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) through the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea, under Grant 21A20131600005, in part by the Basic Science Research Program through the National Research Foundation of Korea through the Ministry of Education under Grant 2016R1D1A1B03933566, in part by the Institute for Information and Communications Technology Promotion grant through the Korea government under Grant 2017-0-00770.

ABSTRACT The traditional big-data analytical approaches use data clustering as small buckets while providing distributed computation among different child nodes. These approaches bring the issues especially concerning network capacity, specialized tools and applications not capable of being trained in a short period. Furthermore, raw data generated through IoT forming big data comes with the capability of producing highly unstructured and heterogeneous form of data. Such form of data grows into challenging task for the real-time analytics. It is highly valuable to have computational values available locally instead of through distributed resources to reduce real-time analytical challenges. This paper proposes a fusion of three different data models like relational, semantical, and big data based data and metadata involving their issues and enhanced capabilities. A case study is used to represent data fusion in action from RDB to Resource Description Framework. Whereas, issues and their feasible solutions are also being discussed in this paper.

INDEX TERMS Big data, data fusion, data transformation, data transformation challenges.

I. INTRODUCTION

Big data comes with characteristics having complexity, autonomous, heterogeneity, and distributed form of an ever growing dataset. These datasets can come from social media, physical sciences, biological and medical sciences [1]. In the history of data science generation history have never been recorded with the capability and capacity of data storage and analytics ever before in the state it is, till now. The challenges arise with the growth of data which are tremendously getting out of the capacity of commonly used software and tools to handle their analytics and management issues [2]. In most of the cases, extracting all information especially in a real-time environment is nearly infeasible. Moreover, currently used methods for handling big data are still incapable [3]. As a result, there is a need for a platform having the competence of providing real-time and quickest prediction response for data analysis.

Big data in real time have diverse and autonomous representations bringing highly unstructured and unrelated data based relationships in producing results which are getting complex and faulty. The heterogeneous data features represent different representations for data. Decrease the effect of heterogeneous and complex data; there can be computationally introduced at localized systems considering they are having better computational power [4]. There can be a way of transforming data into a common data fusion. As a result, the common forms of data in consequence to data fusion will be highly compatible for data linkage and relativity indexing for getting better analytical outcomes [5], [6]. Major of data is stored either using relational, semantical or big data formats. Relational data is stored in the form records containing a collection of singleton cells representing fields supported by its data structure and constraints for an entity. Furthermore, semantical data representation involves triples

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

relating data resource as a subject with the object through a predicate. A triple is three values pair as subject, predicate, and object in generalized which can further be classified for different data representations [7]. At the big data, variety is represented using JSON or JavaScript Object Notation in a key/value pairs. These key/value pairs can further be made in the form of collections or list representing configurations and data with minimal structural representations involved. After the brief introduction of records, triples, and key/value pairs the impact of data and variations, in general, can be visualized clearly. The concept brings a huge need for having a fusion or common platform to bring all varieties of data in a common form for computations to become localized and real-time possible having better accuracy chances [8].

The organization of this manuscript is further divided into five sections. The first section is about the literature review of different data model's used for data fusion as their historical evaluation, impact, and applications. Next section is about the methodology for semantic annotation for big data followed by its implementation. Remaining sections are on results, discussion and conclusion of the manuscript.

II. LITERATURE REVIEW

To understand the data fusion firstly data model and their capability are presented along with their historical evolution. This way, it is easier to understand their limit and importance in the literature.

A. STRUCTURE DRIVEN DATA

The relational data model was first invented with the term "relational database" by E. F. Codd from IBM in 1970. Whereas, Codd had defined relational in his paper titled "A Relational Model of Data for Large Shared Data Banks" in which he had introduced 12 rules for implementing relational data model also known as Codd's rules. These rules were completely taken but up to a minimum and necessary level in defining a table as a relation and operators used to manipulate this data form. Whereas, a language was introduced for querying by Chamberlin and Boyce in 1974 from IBM. It was first named a SEQUEL (Structured English Query Language) which was made standard in ANSI X3H2 committee with SQL (Structured Query Language) in 1986 [9]. In 1976 a designing model to view relational data with the entity-relational model by Peter Chan. In 1990's third generation database system manifesto was introduced by Stonebraker in 1990 which in 1996 became ORDBMS (Object Relational Databases Management System) [10]. Time-wise description of information related to RDB data model history evolution is given in Table 1.

Further history of RDB is concerned with the management system of the relational model [11], [12]. In Fig. 1, the evolution of RDB data and querying model linking them together according to the timeline at the side to show their arrival according to the history using year and author details. Now in next section evaluation of XML is being represented [13], [14].

TABLE 1. RDB history evolution.

Year	Author	Description	Key Term
Jun. 1970	[Codd, Jun. 1970]	A Relational Model for Large Shared Databanks	Relational Data-Model
1974	[Chamberlain, et al., 1974]	SEQUEL: A Structured English Query Language	SEQUEL
1976	[Peter Chen, 1976]	The entity-relationship model—toward a unified view of data	Entity-relationship (ER) model
1986	[ANSI X3H2 committee, 1986]	Became an ANSI standard	SQL
Sep. 1990	[Stonebraker, et al., Sep. 1990]	Third-generation Database System Manifesto	ORDBMS

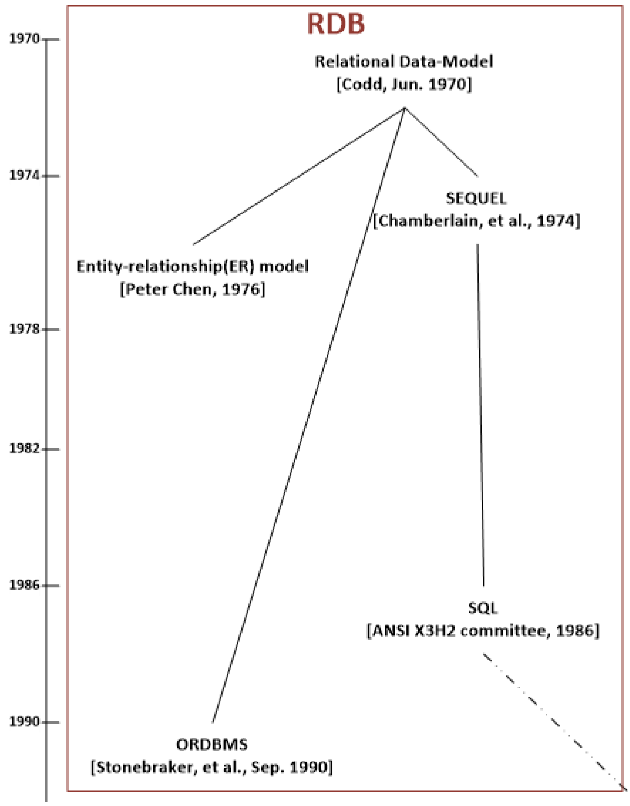


FIGURE 1. RDB history using year and author details.

B. SIMPLE INFERENTIAL DATA

History of XML starts in 1996 with the introduction of XML version 1.0 by Tim Bray. In 1998 Extensible Stylesheet Language (XSL) version 1.0 first was introduced by James Clark et al. and then came the XML Path Language (XPath) version 1.0 in 1999. XPath is used to navigate through XML document object. XSLT version 1.0 and XPath version 1.0 became recommendations in 1999 as a standard. In 2000 and 2001 came XML Schema Part 0: Primer and XQuery (A Query Language for XML) version 1.0 accordingly. In 2001 XML Schema was made as a recommendation. Then updates to XML, XSLT, and XPath were introduced in 2001 [15]. Moreover, update to XML Schema were introduced in 2004. In 2006 XML version 1.0 and 1.1 both were made as re XSLT-based change is a good outline of syntactic change of XML records. For our circumstance, XSLT is used for phonetic change between different XML-designs, e.g., XPath expressions are moreover a possible course of action. Each report of a standard is changed into a looking at an acknowledged structure during the phonetic change stage [16]. XPath is a language used to pinpoint exact XML nodes in a DOM succinctly. Moreover, XQuery is a superset of XPath that also provides FLWOR syntax, which is SQL-like commendations [17]. This history continues, and details can be seen according to the XML data model history evolution separately showing when each concept and their updates were introduced and when they were made standardized [18]. In the start, an idea of a structure capable of being utilized was introduced, and when it was mature, then it became a recommendation to be used as a standard. Evaluation of XML is shown by their recommendations of W3C by linking them together through lines according to the timeline to show their arrival year-wise according to the history [19], [20].

C. JSON (JAVASCRIPT OBJECT NOTATION)

JSON is built on the array and objects purely designed for representing data for ease of interoperability. JSON was publicly introduced by Douglas in 2002. An array of JSON is represented using '[' and ']' brackets whereas an object is represented using "and" bracket. The smallest unit of data is represented in JSON using key/value pair separated using a colon in between them [21].

D. RULE-BASED INFERENTIAL DATA

The Resource Description Framework (RDF) belongs to the family of specifications devised by W3C. The W3C organization had been working on introducing linked data and making it used as a standard for data to be represented in the form of triples. Attractive features of RDF include merging of data even if inner schemas differ and it has change adaptive nature [22]. RDF is now commonly used for representing web resources in the form of conceptual description, and it has its utilization in knowledge management systems [23]. Whereas RDF Schema helps in building a foundation for RDF metadata and it also provides interoperability between different and distributed systems concerning data representation and understandability for machines [24]–[26].

History of RDF Schema specifications used for defining RDF structure having semantics was introduced in March 1999. Whereas, data representation in RDF form known as RDF Primer was introduced in March 2002. On July 2002, feature synopsis for OWL Lite and OWL were introduced. RDF Schema 1.0, RDF Primer, and OWL were made standard in February 2004. SPARQL Query Language for RDF, and RDFa in XHTML syntax and processing rules for embedding RDF through attributes, these both were introduced in October 2004 and 2007 respectively. Whereas, these were made standard in January 2008 and October 2008 respectively. OWL 2 Prime modified form of OWL was introduced in April 2008 and was made standard in October 2009. Similarly, advancements were made in SPARQL as SPARQL 1.1 in October 2009 and were made standard in March 2013. RDF was modified in August 2011 as RDF 1.1 and RDFa was made standard in June 2011. RDB to RDF mapping language R2RML was introduced in October 2010 and was made as a standard language of mapping in September 2012. Modifications in these concepts, rules, and standards are still going through the process of improvements [27], [28]. Time-wise description of information related to RDF data model history evolution was separately showing when each concept and their updates were introduced and when they were made standardized [29], [30]. In the start, an idea of a structure capable of being utilized was introduced, and when it was mature, then it became a recommendation to be used as a standard [31], [32].

Evaluation of RDF is shown by their recommendations of W3C by linking them together through lines according to the timeline at the side of the figure to show their arrival year-wise according to the history. W3C RDF Working Group is continuously working on improvements concerned with RDF data model [33].

E. TOOLS AND APPLICATIONS

Among mapping languages brief introduction of each start with Direct Mapping which provides a direct mechanism to transform RDBs into Semantic Web by mapping table as class and field to properties. Whereas, URIs are generated automatically following RDB schema and data. R2O is aimed to cope complex mapping and low similarities among RDB to ontologies with schematic implementation either found in RDFS or OWL. In Relational.OWL [33], OWL Full based ontology representation to describe the schema and data of an RDB. OpenLink Software a server named Virtuoso Universal Server provides RDF Views to represent relational data on the Semantic Web. A SQL SELECT query is used to translate dataset found in the database into a set of triples. Whereas, SQL DDL forms a syntax level aspect of view. D2RQ [88], [89] is used to transform RDB based data into virtual RDF graphs [34].

Access to the Semantic Web data is through SPARQL queries and Linked Data. It is the descendant of the XML-oriented D2R mapping [20]. Triplify is a query-oriented transformation of RDB into RDF statements to distribute Linked Data from RDBs. Triplify transformation is developed using PHP scripts/code. R2RML a mapping language made a recommendation by W3C to make a standardized approach for RDB to RDF transformation. OntoAccess mediation platform based transformation language known as R3M. As an update, attentive transformation language, it enables providing partial bidirectional query oriented RDF-oriented contact to the RDB. Next in this subsection different slightly modifications and tools are discussed to cover the latest research on the transformation process [35]. Sheet2RDF (2015) is a tool which transforms data available in the spreadsheet into RDF triples following the mechanism of Direct Mapping. It fails to come up with the solution for schema generation. Ultrawrap Mapper (2015) is a tool built-up following semi-automation in mapping RDB to RDF using R2RML language. RDF(S)-OWL (2016) follows Direct Mapping and R2RML for transforming RDB Schema and data to ontology. This work is majorly focused on mapping rules concerning procedure of transformation [36], [37]. SPARQL2X Query interoperability framework (2015) XML, OWL, and RDF.

F. DATA FUSION AND TRANSFORMATION

Update issue is concerned with both sides RDB and RDF data when needed to be updated. Data Reuse feature concerns with XML generated data which further can be used by any source [38]. Whereas, D2RQ/Update only supports write-only at RDF Store but not on RDB. In Table 2.4, features like relation to class, update, record URI, data reuse, datatypes, integrity constraints, write support, data transformation, query base transformation, and bidirectional transformation are mapped [39]. It is now clear that bidirectional transformation, update, and write support features necessarily required for updating a data and schema of either data model of RDB or RDF are in R3M only which is again query oriented and partially supported [40]. No other approach for transformation provides the required skill set to accomplish bi-directional data transformation with improved capability and capacity to solve update issue. Where update issue is about a change introduced in data either available at RDF or RDB should also be updated only at the point where it appears in RDF if data is changed in RDB whereas in RDB if data is changed in RDF. This study works its way in resolving the issue of update by introducing a mapping mechanism in the common intermediate form of data gained through data transformation. There are API's available in JavaScript platform which can easily transform XML into JSON data pairs [27].

III. METHODOLOGY FOR SEMANTIC ANNOTATION FOR BIG DATA

Real-time data collection is found mostly in the form of sensors data collected through physical or biological resources. In the current era of information analytics Internet of Things (IoT) is playing the main role in managing, controlling and monitoring of the resources even at remote locations. With the involvement of social medium and mobile communication data is increasing rapidly. At the end of big data, Hadoop is playing a key role through its platform in data collection, computational clustering of distributed units, and dramatic fast analytics. However, still, it lacks in real-time boosted analytics for a localized fast outcome. For that to work data fusion is proposed at the level of localized or short area cluster of units to have highly interactive

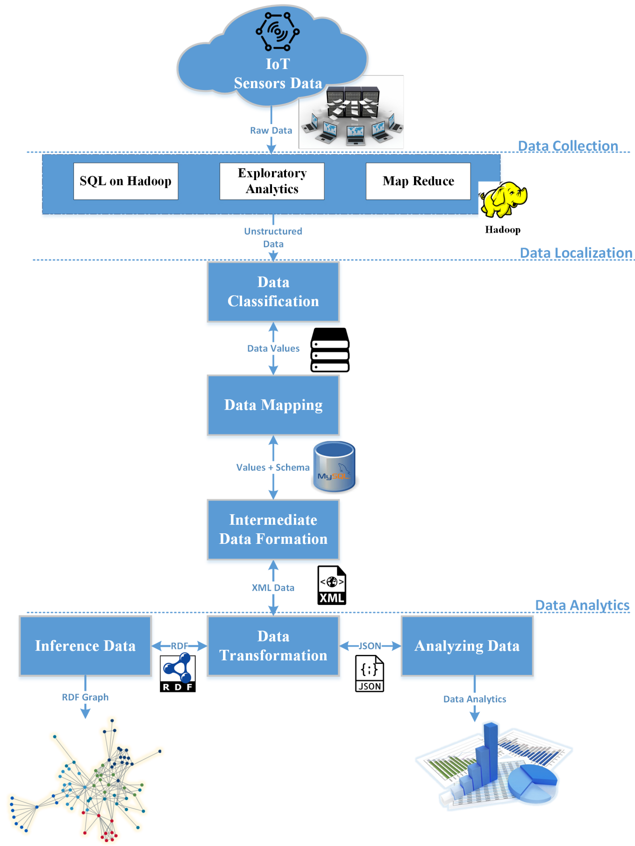


FIGURE 2. Research Methodology for semantic annotation for big data.

data transformation platform. Data found either through big data, relational DB or RDF/OWL is further transformed into the needed shape and form following that needed form is XML, JSON, or RDF.

The methodology started from raw data collected from sensors and transferred directly to Hadoop. Followed by data is also transformed into XML as intermediate data format. Then this is further fused in the format of JSON (highly interoperable) and RDF (for data linkage and inferencing) as shown in Fig. 2. The main component of this methodology is the data mapping where differences are identified and solved. Once data is transformed into the linked formate and JSON then it is ready for data analytics and inference side by side.

A. UNDERSTANDING CHALLENGES

The challenges involved in the methodology for real-time data fusion for localized big data's analytics concerns with data updates. Other issues involve one data model support and limitation to other data model during the process of data fusion. Data collected in traditional data storage representing relation database where data is placed separately from metadata. The new generation data formats like, JSON and RDF are more data and hierarchy oriented. So, XML needs to map all loopholes of one data model to other for covering any possible data fusion issues (as shown in Fig. 3). Here comes the reason that is freedom found in XML makes data fusion of heterogeneous possible and real. On the other end, JSON comes with the similar feature, but it only focuses on the data leaving the structure part data alone.

IV. DATA FUSION IMPLEMENTATION

In this section, data fusion implementation is covered using four algorithms of data transformation from RDB to RDF. In Algorithm 1 focus is on the transformation from RDB to XML. In this algorithm, it takes RDB document as input and produces XML document at the end of its execution as output. This algorithm starts at line 2 with assigning a header by defining XML version used an encoding which is UTF-8. At line 3 concerned DB is being selected and at line 4 XML document is given name equivalent to the RDB name. At line 5 an RDB tag has been started having attributes like names space and location of XML schema identifications. From line 6 up to line 19 information related to each relations instance is being extracted from RDB and converted into XML tag representation as a sub-element. These sub-elements become part of elements of relation they belong to an XML document. In the end, full RDB relation's records are being transformed into XML tags.

Algorithm 1 Transformation from RDB to XML
Input: RDB Document
Output: XML Document (XML Values)

1. ***Begin***
2. ***Build*** *Tag <?xml version="1.0" encoding= "UTF-8"?>*
3. ***Select*** *RDB from the document*
4. ***Make*** *XML Document.name as RDB.name*
5. ***Build*** *Tag <RDB.name xmlns:xs = "http :// www. w3.org /2001/ XML Schema "xmlns :xsi=" http : // www. w3. Org /2001 /XMLSchema-instance" xsi:schemaLocation="http://www.w3schools.com XSD output RDBS.name.xsd">*
6. *Suppose RDB has total n tables in it*
7. *Loop* ***For*** *i = 1 to n do*
8. *Select* $table_i$textit.name from RDB
9. *Suppose* $table_i$*.name has total m fields in it*
10. *Suppose* $table_i$ *.name has total k records in it*
11. *Inner Loop For l = 1 to k do*
12. *Build Tag <*$table_i$*.name>*
13. *Inner Loop For j = 1 to m do*
14. *Select* $field_j$*.name and* $field.value_{j,l}$ *of* $table_i$
15. *Build Tag <*$field_j$*.name>* $field.\ value_{j,l}$ *< /*$field_j$*.name >*
16. *End Inner Loop*
17. *Build Tag </*$table_i$*.name>*
18. *End Inner Loop*
19. ***End*** *Loop*
20. ***Build*** *Tag </RDB.name>*
21. ***End***

In Algorithm 2, there is two type of triples are built. One containing DB field and literal for representing the value of that field from child elements. Second, is to link these fields and values to a table or relation from parent elements.

Algorithm 2 Transformation from XML to RDF
Input: XML Document (XML Values)
Output: RDF Triples

1. ***Begin***
2. ***Select*** *XML Document.name from the document*
3. ***Build*** *Triple XML Document.name rdfs:Class rdf:resource*
4. */∗ here dot symbol shows property of the document selected∗/*
5. *Suppose XML Document has total n parent elements for relations in it*
6. *Suppose XML Document has total m sub elements within each parent element*
7. *Loop* ***For*** *i = 1 to n do*
8. *Inner Loop* ***For*** *j = 1 to m do*
9. *Select sub-*$element_j$*of* $element_i$
10. *Make Triple* $element_i$*.name sub-*$element_j$*.name sub-*$element_j$*.value*
11. *End Inner Loop*
12. *End Loop*
13. ***End***

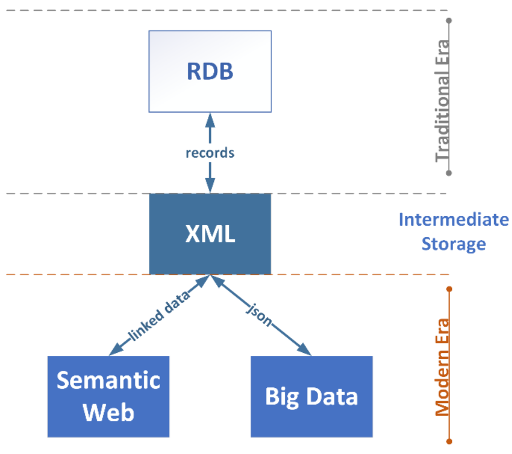


FIGURE 3. Research Methodology for semantic annotation for big data.

these fields and values to a table or relation from parent elements.

In Algorithm 3, there is two type of tags are built. One works as a parent tag extracted from a triple of type class. These tags are relations of RDB. The second one is child tags extracted properties for values and constraints.

In Algorithm 4, there is two type of triples are built. One containing DB field and literal for representing the value of that field from child elements. Second, is to link these fields and values to a table or relation from parent elements.

In Fig. 4, a bar chart is shown using some elements calculated relation-wise instances during the data transformation process. These values are represented as CSV values with blue color bar. Color-wise legends in given chart are showing some records, tags, triples generated during the transformation process phase for data as an outcome.

Algorithm 3 Transformation from RDF to XML
Input: RDF Document (RDF/XML Syntax File)
Output: XML Document (XML Values)

1. ***Begin***
2. ***Build*** *Tag <?xml version="1.0" encoding="UTF-8"?>*
3. ***Select*** *RDB from the document*
4. ***Make*** *XML Document.name as RDB.name where xmlns:RDB.name**
5. ***Build*** *Tag <RDB.name xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3schools.com XSD output RDBS.name.xsd">*
6. ***Suppose*** *RDF has total n distinct description tags representing each relation*
7. *Loop* ***For*** *i = 1 to n do*
8. *After removing URI string up-to # symbol RDF tag*$_i$*. name achieved from RDF*
9. ***Select*** *RDF tag*$_i$*.name from RDF*
10. ***Suppose*** *RDF tag*$_i$ *has total m RDF sub tag in it*
11. ***Suppose*** *RDF tag*$_i$*.name has total k occurrences in RDF*
12. *Inner Loop* ***For*** *l = 1 to k do*
13. *Build Tag <RDF tag*$_i$*.name>*
14. *Inner Loop* ***For*** *j = 1 to m do*
15. *Select RDF sub tag*$_j$*.name and RDF sub tag*$_{j,l}$*.value of RDF tag*$_i$
16. *Build Tag <RDF sub tag*$_j$*.name>RDF sub tag*$_{j,l}$*.value </RDF sub tag*$_j$*.name>*
17. ***End*** *Inner Loop*
18. *Build Tag </RDF tag*$_i$*.name>*
19. ***End*** *Inner Loop*
20. ***End*** *Loop*
21. *Build Tag </RDB.name>*
22. ***End***

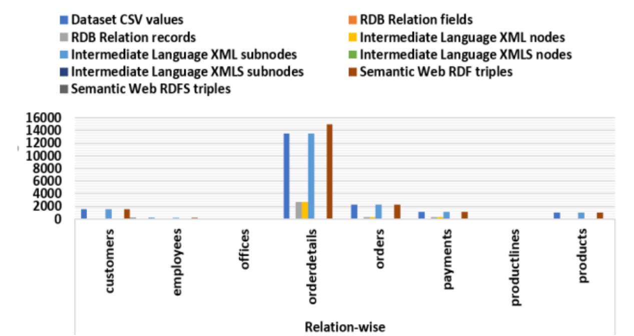


FIGURE 4. Graph representing summarizing values vs. relations for each data form.

Algorithm 4 Transformation from XML to RDB
Input: XML Document (XML Values)
Output: RDB

1. ***Begin***
2. ***Select*** *XML Document.name from the document*
3. ***Make*** *RDB.name equal to XML Document.name*
4. */* here dot symbol shows property of the document selected*/*
5. ***Suppose*** *XML Document has total n distinct parent elements representing each relation*
6. ***Suppose*** *XML Document has total k occurrences of each parent element*
7. ***Suppose*** *XML Document has total m sub elements within each parent element*
8. *Loop* ***For*** *i = 1 to n do*
9. *Inner Loop For j = 1 to k do*
10. ***Select*** *sub-element*$_j$ *of element*$_i$
11. *Query String = "'INSERT INTO element*$_i$*.name ("'*
12. *Inner Loop* ***For*** *l = 1 to m-1 do*
13. *+"'sub-element*$_l$*.name,"'*
14. *End Inner Loop*
15. *+"'sub-element*$_l$*.name)"'*
16. *+"' VALUES ("'*
17. *Inner Loop* ***For*** *l = 1 to m-1 do*
18. *+"'sub-element*$_l$*.value,"'*
19. *End Inner Loop*
20. *+"'sub-element*$_l$*.value)"'*
21. *End Inner Loop*
22. *Execute Query(Query String)*
23. *End Loop*
24. ***End***

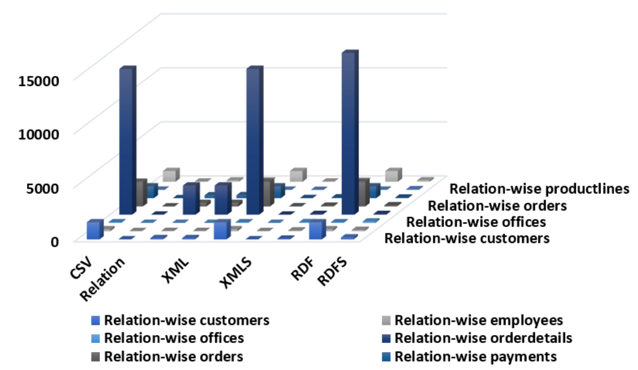


FIGURE 5. Bar graph of summarized DB transformation result.

In Fig. 5, a bar graph of summarized RDB after passing data transformation phases are shown in 3D bars. The results are shown in graph depicts some triples are quite high compared to original values. Its better due to this will ensure better linkage capabilities among the data in RDF for referencing and querying using SPARQL.

TABLE 2. "class models" RDB data transformation summary.

Type	Representat ion	Relation-wise							
		orderdet ails	employ ees	offic es	orde rs	payme nts	productli nes	custom ers	produ cts
CSV	values	13495	184	63	2282	1092	28	1586	990
Relati on	fields	5	8	9	7	4	4	13	9
	records	2699	23	7	326	273	7	122	110
XML	nodes	2700	24	8	327	274	8	123	111
	subnodes	13495	184	63	2282	1092	28	1586	990
XML S	nodes	6	9	10	8	5	5	14	10
	subnodes	70	70	70	70	70	70	70	70
RDF	triples	14980	184	63	2282	1092	28	1586	990
RDFS	triples	69	121	135	107	56	61	183	119

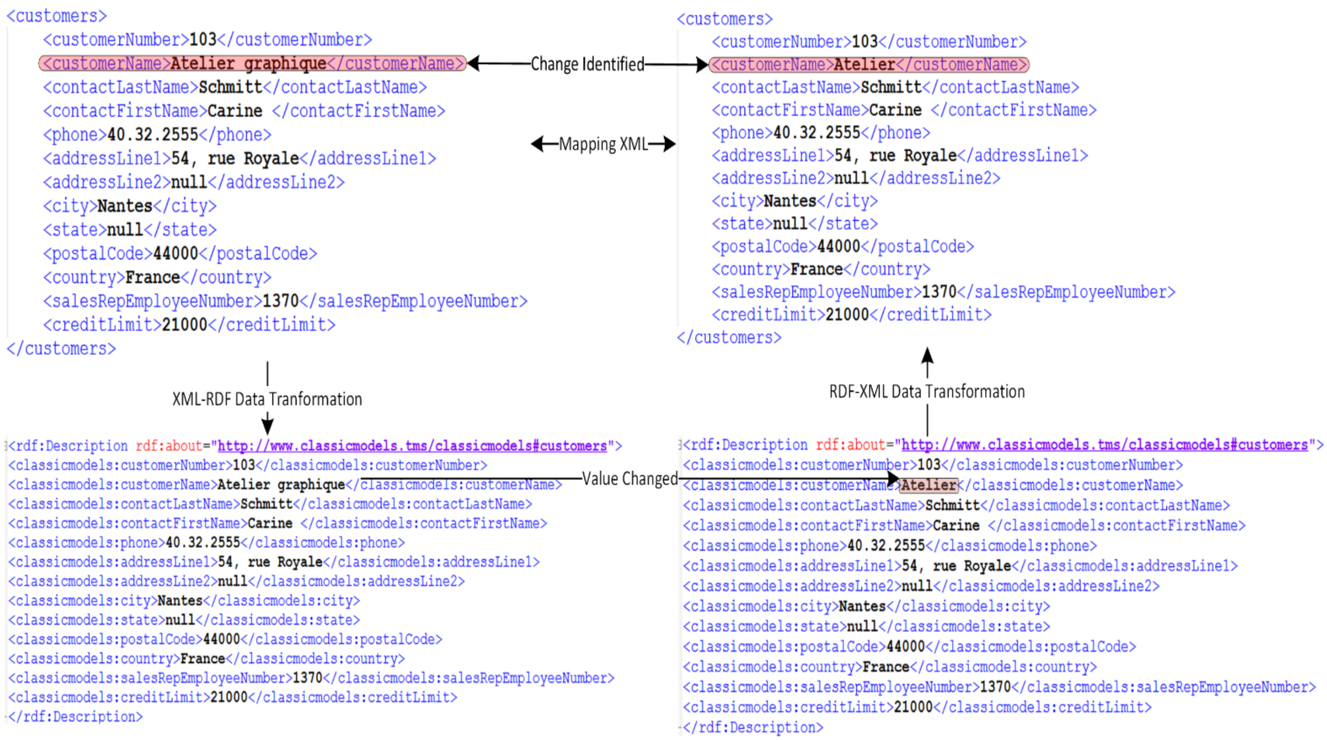


FIGURE 6. Impact of value changed induced into RDF data when mapping XML files.

V. DATA FUSION RESULTS

A case study based on a relational database freely available online in the form of "ClassicModels" is used to test the data fusion implementation between RDB, XML, RDF, and JSON. The relation has been passed through two different phases of data fusion. Among these two fusion phases, one is from RDB to XML, and then into RDF. Resultant data generated after the transformation is tested and verified by following W3C standards of data representations RDF. In Table 2, each relation-wise summary of data transformation into RDB, XML, and RDF have been presented. This summary starts with the dataset 19720 values in total. Here relations like order details, orders, customers, employees, payments, product lines, offices, and products share 1586, 184, 63, 13495, 2282, 1092, 28, and 990 values each accordingly. Now in XML, we have two formats one is the schema and other is nodes for value representation. There are two node tags along with sub-node tags of elements in XML for representing each relation in RDB. Moreover, there are 3575 nodes in total for the capturing information transformed from RDB. Generated triples for RDF remain same in number as for each sub-node tags found in XML.

VI. DISCUSSION

One feature of big data is to work with a variety of data, which can be in any form coming into or going out of

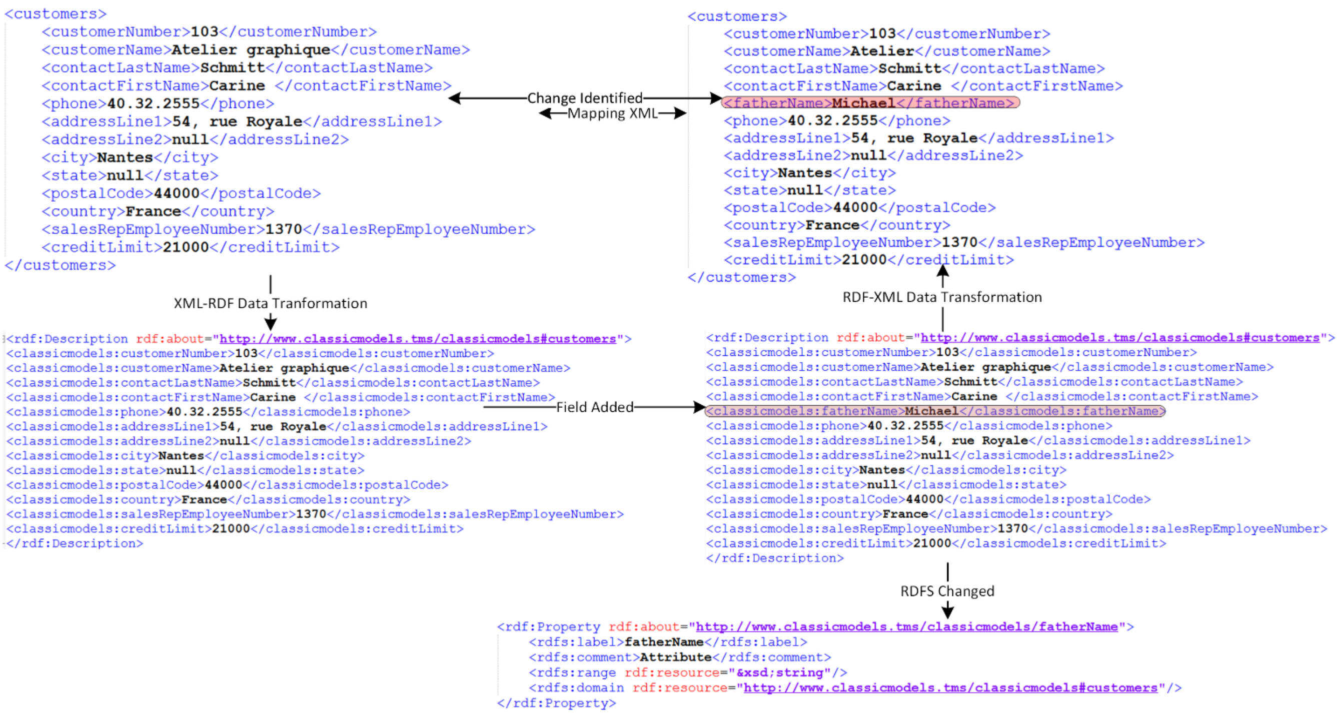
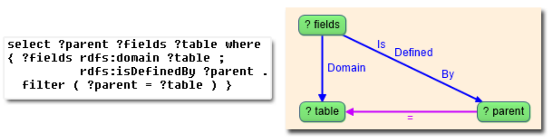


FIGURE 7. **Impact of field added to RDF data when mapping XML files.**



(a) (b)

FIGURE 8. **Primary Key based Triple Extraction (a) SPARQL query for primary keys extraction (b) Graphical representation of the query.**

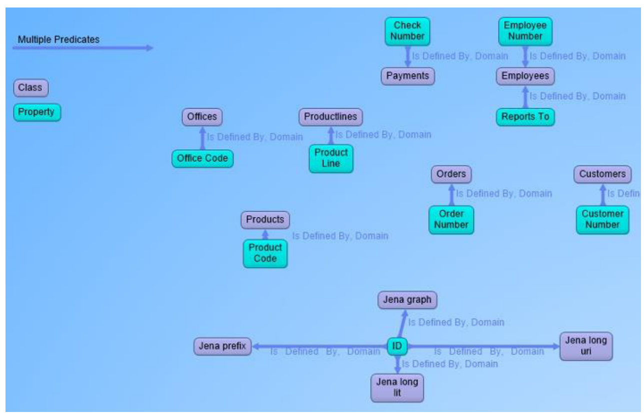


FIGURE 9. **RDF Graph for a query of primary keys.**

the system. This data can be semantically rich data, relational data, hierarchical data, or another form of data. Therefore, data found in the shape of RDF, RDB or XML needs to be capable of transforming in any direction. Which is yet in the form of maturing as a problem, which needs to be addressed and sorted out. Our study focus is on data fusion of heterogeneous data into RDF or JSON. As a special case, we have focused on RDB and RDF based on data transformation. Afterward, focus on a data fusion to work with our generalized proposed model of data capable of translating each data category into XML form. Passing through transformation algorithms, it results in more capable of further production of analytical analysis. The change produced in RDF Store can easily be measured by mapping XML files one came from RDB and the other one achieved from RDF file gained by passing through the data-transformation process. Change can be from one of the following cases along with solution set:

Case 1: Changing the value of a records item (as shown in Fig. 6)

Solution. 01: Update the DB record

Case 2: Adding new element in the record previously none existed (as shown in Fig. 7)

Solution.01: Update the XSD file by adding a new field in meta-data

Solution.02: Also, add it to DB table as a new field

Case 3: Directly changing the structure of the RDFS file

Solution.01: Update the XSD file by adding updates to meta-data

Solution.02: Also, add change to DB table

As data is found in all forms like RDF, XML, and RDF, which makes it much more useful in al kind of machinery, and application, which are major, concern with data itself. Modification performed at any data form of RDF, XML or RDF can be reflected easily by the represented method of transformation which also opens a new window of improvement in the field of semantic web. This way its utilization can be maximized by involving any data available.

A. PRIMARY KEY BASED TRIPLE EXTRACTION

By completing one directional transformation, it is necessary to extract all information from RDF store. Simple SPARQL queries do support information needed, but the toughest of all is the extraction and tagging of the constraint related information. The gruff tool further helps in analyzing optimized query for the primary key-based information extraction. This query and its graphical representation are shown in Fig. 8. Whereas, this query resultant triples and elements after the execution of the query are represented in Fig. 9.

B. FOREIGN KEYS AND COMPOSITE KEYS BASED TRIPLE EXTRACTION

Similarly, the gruff tool further helps in analyzing optimized query for the primary key-based information extraction. This query and its graphical representation are shown in Fig. 10. Whereas, this query resultant triples and elements after the execution of the query are shown in Fig. 11.

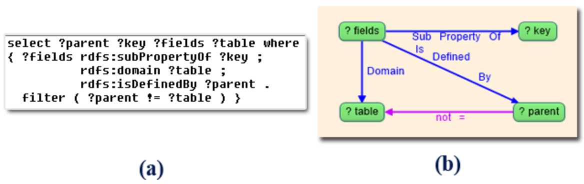


FIGURE 10. Foreign Key based Triple Extraction (a) SPARQL query for foreign keys extraction (b) Graphical representation of the query.

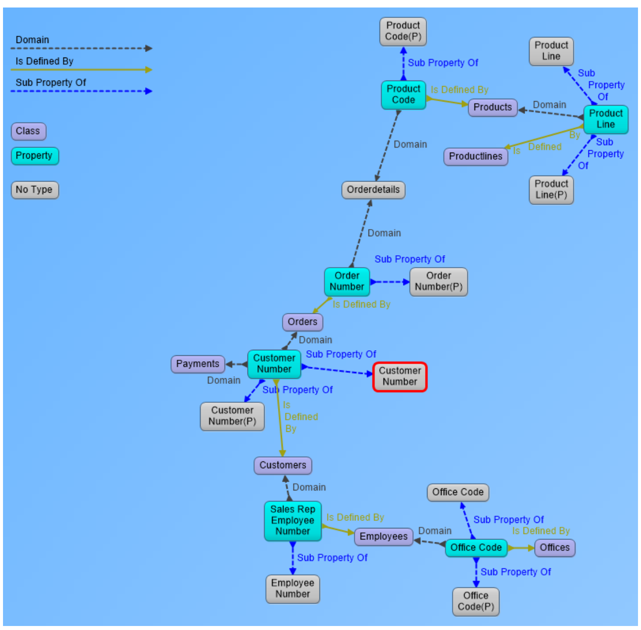


FIGURE 11. RDF Graph for a query of foreign keys.

VII. CONCLUSION

Now by looking at the result and discussion section of this manuscript where the methodology is being tested and analyzed. This study shows all feature necessary for data fusion to become localized and feasible to support change oriented update for data and metadata. The only feature which is showing partial support for query oriented transformation is not a bad thing due to the limitation of query concerned with the management of data and metadata. Because SPARQL, XQuery and SQL query platforms used for RDF, XML, and RDB accordingly are only used to navigate data and metadata which was necessarily needed for data fusion to happen. This method also represents that our presented model is much richer on data transformation to have even bidirectional transformation support along with features like data reusability, write and update support for data and metadata. Whereas, other languages and platforms are in lacking to help in resolving update of data and metadata in either whole system or a specific part of the system. Bidirectional data transformation causes complete data and metadata to be generated in case of any change introduced in either RDB or RDF sides of the data store. Now considering the high adaptability of data in the local space of storage is ready for big data related analytics are easily capable for the real-time computations.

REFERENCES

[1] P. Y. Vandenbussche, G. A. Atemezing, M. Poveda-Villalón, and B. Vatant, "Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web," *Semantic Web*, vol. 8, no. 3, pp. 437–452, 2017.

[2] K. Höffner, "Survey on challenges of question answering in the semantic Web," *Semantic Web*, vol. 8, no. 6, pp. 895–920, 2017.

[3] M. Nentwig, M. Hartung, A.-C. N. Ngomo, and E. Rahm, "A survey of current link discovery frameworks," *Semantic Web*, vol. 8, no. 3, pp. 419–436, 2017.

[4] A. Gangemi, V. Presutti, D. R. Recupero, A. G. Nuzzolese, F. Draicchio, and M. Mongiovì, "Semantic Web machine reading with FRED," *Semantic Web*, vol. 8, no. 6, pp. 873–893, 2017.

[5] I. R. Goodman, R. P. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, vol. 37. Dordrecht, The Netherlands: Springer, 2013.

[6] M. Bevilacqua, A. Tsourdos, A. Starr, and I. Durazo-Cardenas, "Data fusion strategy for precise vehicle location for intelligent self-aware maintenance systems," in *Proc. IEEE 6th Int. Conf. Intell. Syst., Modelling, Simulation (ISMS)*, Feb. 2015, pp. 76–81.

[7] A. Hotho, R. Jäschke, and K. Lerman, "Mining social semantics on the social Web," *Semantic Web*, vol. 8, no. 5, pp. 623–624, 2017.

[8] D. Calvanese *et al.*, "Ontop: Answering SPARQL queries over relational databases," *Semantic Web*, vol. 8, no. 3, pp. 471–487, 2017.

[9] R. Mukherjee, "Interfacing data destinations and visualizations: A history of database literacy," *New Media Soc.*, vol. 16, no. 1, pp. 110–128, 2014.

[10] G. Palmer, P. A. Stephens, A. I. Ward, and S. G. Willis, "Nationwide trophic cascades: Changes in avian community structure driven by ungulates," *Sci. Rep.*, vol. 5, Oct. 2015, Art. no. 15601.

[11] A. Grimm, "Life-history trait database of European reptile species," *Nature Conservation*, vol. 9, pp. 45–67, Dec. 2014.

[12] C. Deschaseaux, P. Sharma, A. Gabrielsen, A. Siadimas, M. Bauer, and R. Przybysz, "Real-world hsCRP testing, with a potential to evaluate residual inflammatory cardiovascular risk, among patients with history of myocardial infarction in the United States: A retrospective database analysis," *Circulation, Cardiovascular Quality Outcomes*, vol. 10, p. A229, Mar. 2017.

[13] U. Czerwinska, L. Calzone, E. Barillot, and A. Zinovyev, "DeDaL: Cytoscape 3 app for producing and morphing data-driven and structure-driven network layouts," *BMC Syst. Biol.*, vol. 9, no. 1, p. 46, 2015.

[14] S. Liu, B. Wang, P. T. Bremer, and V. Pascucci, "Distortion-guided structure-driven interactive exploration of high-dimensional data," in *Computer Graphics Forum*. New York, NY, USA: Wiley, 2014.

[15] A. Patwardhan and R. Patwardhan. (Jun. 2016). "XML entity architecture for efficient software integration." [Online]. Available: https://arxiv.org/abs/1606.07941

[16] R. Cusack *et al.*, "Automatic analysis (aa): Efficient neuroimaging workflows and parallel processing using MATLAB and XML," *Frontiers Neuroinform.*, vol. 8, p. 90, Jan. 2015.
[17] G. Post *et al.*, "XHSTT: An XML archive for high school timetabling problems in different countries," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 295–301, 2014.
[18] X. M. L. Lemaire and D. Kerr, "Sustainable thermal energy service partnerships (STEPs) blog," UCL Energy Inst., London, U.K., Tech. Rep., 2014.
[19] W. L. Bramlett, Jr., and B. C. Giedt, "System and method for embedding report descriptors into an XML string to assure report consistency," U.S. Patent 9 471 892, Oct. 18, 2016.
[20] L. Xiyin, C. Lijun, and Z. Zhang, "The study of XML functional dependency and multi-valued dependency and inference rules set," *Int. J. Database Theory Appl.*, vol. 8, no. 6, pp. 343–354, 2015.
[21] M. C. Munro, "Text functions (JSON)," in *Learn FileMaker Pro*, vol. 16. Berkeley, CA, USA: Springer, 2017, pp. 313–320.
[22] I. Dongo, Y Cardinale, F Al-Khalil, R Chbeir, "Semantic Web datatype inference: Towards better RDF matching," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2017, pp. 57–74.
[23] K. M. Nunes, M. V. O. Andrade, A. M. S. Filho, M. C. Lasmar, and M. M. Sena, "Detection and characterisation of frauds in bovine meat in natura by non-meat ingredient additions using data fusion of chemical parameters and ATR-FTIR spectroscopy," *Food Chem.*, vol. 205, pp. 14–22, Aug. 2016.
[24] J. H. Carpenter, A. C. Robinson, B. J. Debusschere, and A. E. Mattsson, "Automated generation of tabular equations of state with uncertainty information," Sandia Nat. Lab., Livermore, CA, USA, Tech. Rep. SAND2015-8809C, 2015.
[25] J. Olsson, T. Pavlenko, and F. L. Rios, "Bayesian inference in decomposable graphical models using sequential Monte Carlo methods," Math. Statist., Sweden, Switzerland, Tech. Rep., 2017.
[26] B. Kimelfeld and P. Senellart, "Probabilistic XML: Models and complexity," in *Advances in Probabilistic Databases for Uncertain Information Management*. Heidelberg, Germany: Springer, 2013, pp. 39–66.
[27] F. Schmidt, "Perception and inference," in *Exploring Alterity in a Globalized World*. New York, NY, USA: Routledge, 2016.
[28] C. Dong, Y. Yuan, and W. Lei, "Additive manufacturing cloud based on multi agent systems and rule inference," in *Proc. IEEE Electron. Autom. Control Conf. Inf. Technol. Netw.*, May 2016, pp. 45–50.
[29] H. Paulheim and C. Bizer, "Type inference on noisy RDF data," in *Proc. Int. Semantic Web Conf.*, 2013, pp. 510–525.
[30] J. Kim and S. Park, "Efficient authorization conflict detection using graph labeling in controlling access to RDF," *Comput. Standards Interfaces*, vol. 38, pp. 51–63, Feb. 2015.
[31] J. Subercaze, C. Gravier, J. Chevalier, and F. Laforest, "Inferray: Fast in-memory RDF inference," *Proc. VLDB Endowment*, vol. 9, no. 6, pp. 468–479, 2016.
[32] J. Kim, "Authorization conflict problems in combining RIF rules with RDF data," *IEICE Trans. Inf. Syst.*, vol. E98.D, no. 4, pp. 863–871, 2015.
[33] J. A. Khan and S. Kumar, "OWL, RDF, RDFS inference derivation using Jena semantic framework & pellet reasoner," in *Proc. IEEE Int. Conf. Adv. Eng. Technol. Res. (ICAETR)*, Aug. 2014, pp. 1–8.
[34] D.-H. Im, S.-W. Lee, and H.-J. Kim, "Backward inference and pruning for RDF change detection using RDBMS," *J. Inf. Sci.*, vol. 39, no. 2, pp. 238–255, 2013.
[35] J. Urbani and C. Jacobs, "RDF-SQ: Mixing parallel and sequential computation for top-down OWL RL inference," in *Proc. Int. Workshop Graph Struct. Knowl. Represent. Reasoning*, 2015, pp. 125–138.
[36] M. B. Alves, C. V. Damásio, and N. Correia, "RDF query and inference in Prolog," in *Proc. Int. Conf. Knowl. Eng. Semantic Web*, 2016, pp. 191–201.
[37] S. Benbernou, X. Huang, and M. Ouziri, "Fusion of big RDF data: A semantic entity resolution and query rewriting-based inference approach," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2015, pp. 300–307.
[38] M.-A. Baazizi, H. B. Lahmar, D. Colazzo, G. Ghelli, and C. Sartiani, "Schema inference for massive json datasets," in *Proc. Extending Database Technol. (EDBT)*, 2017, pp. 222–233.
[39] T. Adam, V. Leos-Barajas, R. Langrock, and F. van Beest, "Using hierarchical hidden Markov models for joint inference at multiple temporal scales," in *Proc. 32nd IWSM*, vol. 2. 2017, pp. 183–186.
[40] J. Montgomery, M. D. Reid, and B. Drake, "Protocols and structures for inference: A RESTful API for machine learning," in *Proc. 2nd Int. Conf. Predictive APIs Apps*, 2016, pp. 1–14.

SOHAIL JABBAR was a Post-Doctoral Researcher with Kyungpook National University, Daegu, South Korea. He also served as an Assistant Professor with the Department of Computer Science, COMSATS Institute of Information Technology (CIIT), Sahiwal, where he also headed the Networks and Communication Research Group. He is currently an Assistant Professor with the Department of Computer Science and the Director of Graduate Schools, Faculty of Sciences, National Textile University, Faisalabad, Pakistan. He has been involved in many National and International Level Projects. He has authored or co-authored one Book, two Book Chapters, and over 65 International Publications. His research work was published in various renowned journals and magazines of the IEEE, Springer, Elsevier, MDPI, and Old City Publication, and conference proceedings of the IEEE, ACM, and IAENG. He received many awards and honors from the Higher Education Commission of Pakistan, Bahria University, CIIT, and the Korean Government. Among those awards, Best Student Research Awards of the Year, the Research Productivity Award, and the BK-21 Plus PostDoctoral Fellowship are few. He received the Research Productivity Award from CIIT in 2014, 2015, and 2016, respectively. He is currently a TPC member/chair in many conferences. He is on collaborative research with renowned research centers and institutes around the globe on various issues in the domains of Internet of Things, wireless sensor networks, and big data. He is a Guest Editor of *Concurrency and Computation Practice and Experience* (Wiley), *Future Generation Computer Systems* (Elsevier), *Peer-to-Peer Networking and Applications* (Springer), the *Journal of Information and Processing System* (KIPS), and *Cyber Physical System* (Taylor & Francis).

KALEEM R. MALIK is currently pursuing the Ph.D. degree in computer science from the University of Engineering and Technology, Lahore, Pakistan. He was a Lecturer with the Department of Software Engineering, Government College University, Faisalabad, Pakistan, from 2013 to 2015. He has been an Assistant Professor with the Department of Computer Science, COMSATS Institute of Information Technology, Sahiwal, since 2015. He was also an Instructor of computer science with the Virtual University of Pakistan for about five years. His areas of research include but are not limited to semantic web, data modeling, data mining, and databases. His research work is published in various renowned journals of Springer and Hindawi. He has been the Reviewer for leading International and National journals (ISI indexed) such as *Journal of Supercomputing* (ISI index: 1.088) and *The Nucleus* Journal (ISI Indexed).

MUDASSAR AHMAD is serving as an Assistant Professor with the Department of Computer Science, National Textile University, Pakistan. He has 17 years' experience as a Network Manager in a Textile Industry. His research work is published in many conferences and journals. His research includes Internet of Things, bid data, and health care. He is an Associate Editor in the IEEE Newsletters.

OMAR ALDABBAS was born in Al-Salt, Jordan, in 1980. He received the B.E. degree from Philadelphia University, Jordan, in 2003, and the M.Sc. and Ph.D. degrees from De Montfort University, Leicester, U.K., in 2006 and 2008, respectively, all in computer engineering. In 2008, he joined the Department of Computer Engineering, Faculty of Engineering, AlBalqa' Applied University, Amman, Jordan, as a Lecturer. In 2009, he turned into an Assistant Professor. From 2013 to 2015, he was the Vice Dean of the Faculty of Engineering, AlBalqa' Applied University. In 2016, he became an Associate Professor. Since 2016, he has been a Director of the Consultations, Studies and Training Center, AlBalqa' Applied University.

MUHAMMAD ASIF received the M.S. and Ph.D. degree from the Asian Institute of Technology (AIT), in 2009 and 2012, respectively, on HEC Foreign Scholarship. He is currently the Chairman with the Department of Computer Science, National Textile University, Faisalabad. Before this, he was a Research Scholar with the Computer Science and Information Management Department, AIT, Thailand. During the course of time, he was a Visiting Researcher with the National Institute of Information Tokyo, Japan. He has worked on some projects including Air Traffic Control System of Pakistan Air force. He is serving as a Reviewer of a number of reputed journals and also authored a number of research papers in reputed journals and conferences. He is also a permanent member of Punjab Public Service Commission as an Advisor and a Program Evaluator with the National Computing Education Accreditation Council, Islamabad.

SHEHZAD KHALID received the degree from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, in 2000, the M.Sc. degree from the National University of Science and Technology, Pakistan, in 2003, and the Ph.D. degree from the University of Manchester, U.K., in 2009. He is currently a Professor and the Head of Department with the Department of Computer Engineering, Bahria University, Pakistan. He is also a qualified Academician and a Researcher with over 60 international publications in various renowned journals and conference proceedings. He is also the Head of Computer Vision and Pattern Recognition Research Group which is a vibrant research group undertaking various funded research projects. His areas of research include but are not limited to shape analysis and recognition, motion based data mining and behavior recognition, medical image analysis, ECG analysis for disease detection, biometrics using fingerprints, vessels patterns of hands/retina of eyes, ECG, Urdu stemmer development, short and long multi-lingual text mining, and Urdu OCR. He was a recipient of the Best Researcher Award for the year 2014 from Bahria University. He was also a recipient of the Letter of Appreciation for Outstanding research contribution in year 2013 and outstanding performance award from 2013 to 2014. He was the Reviewer for various leading ISI indexed journals.

KIJUN HAN received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 1979, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, South Korea, in 1981, and the M.S. and Ph.D. degrees in computer engineering from the University of Arizona, in 1985 and 1987, respectively. He served as a Researcher of the Agency for Defense Development from 1981 to 1984. He has been a Professor with the School of Computer Science and Engineering, Kyungpook National University, South Korea, since 1988. He headed good number of National and International Level Projects. His research work is published in various renowned journals and magazines of the IEEE, Springer, Elsevier, MDPI, Old City Publication and Hindawi, and conference proceedings of the IEEE and ACM. His research interests include Internet of Things, wireless sensor networks, and big data.

SYED HASSAN AHMED (S'13–M'17–SM'18) received the B.S degree in computer science from the Kohat University of Science and Technology, Pakistan, and the combined master's/Ph.D. degree from the School of Computer Science and Engineering (SCSE), Kyungpook National University (KNU), South Korea. In 2015, he was a Visiting Researcher with Georgia Tech, Atlanta, USA. He is currently a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, USA. He has authored or co-authored over 100 international publications, including Journal articles, Conference Proceedings, Book Chapters, and two books. His current research interests include sensor and ad hoc networks, cyber-physical systems, vehicular communications, and future Internet. From 2014 to 2016, he consequently received Research Contribution Awards by SCSE, KNU. In 2016, his work on robust content retrieval in future vehicular networks leads him to receive the Qualcomm Innovation Award from KNU.