sustainability

Article

Data Compatibility to Enhance Sustainable Capabilities for Autonomous Analytics in IoT

Kaleem Razzaq Malik [1,2], Masood Habib [2], Shehzad Khalid [3], Farhan Ullah [2], Muhammad Umar [2], Taimur Sajjad [2] and Awais Ahmad [4,*]

[1] Department of Computer Science and Engineering, University of Engineering and Technology, Lahore 54890, Pakistan; krmalik@ciitsahiwal.edu.pk

[2] Department of Computer Science, COMSATS Institute of Information Technology, Sahiwal 57000, Pakistan; masoodhabib@ciitsahiwal.edu.pk (M.H.); farhanmscs@ciitsahiwal.edu.pk (F.U.); mumar@ciitsahiwal.edu.pk (M.U.); taimursajjad@ciitsahiwal.edu.pk (T.S.)

[3] Department of Computer Engineering, Bahria University, Islamabad 44000, Pakistan; shehzad@bahria.edu.pk

[4] Department of Information and Communication Engineering, Yeungnum University, Gyeongsan 38541, Korea

* Correspondence: awais@knu.ac.kr; Tel.: +82-104-365-1029

Academic Editor: Seungmin Rho
Received: 18 April 2017; Accepted: 15 May 2017; Published: 23 May 2017

Abstract: The collection of raw data is based on sensors embedded in devices or the environment for real-time data extraction. Nowadays, the Internet of Things (IoT) environment is used to support autonomous data collection by reducing human involvement. It is hard to analyze such data, especially when working with the sensors in a real-time environment. On using data analytics in IoT with the help of RDF, many issues can be resolved. Resultant data will have a better chance of quality analytics by reforming data into the semantical annotation. Industrial correspondence through data will be improved by the induction of semantics at large due to efficient data capturing, whereas one popular medium of sensors' data storage is Relational Database (RDB). This study provides a complete automated mechanism to transform from loosely structured data stored in RDB into RDF. These data are obtained from sensors in semantically annotated RDF stores. The given study comprises methodology for improving compatibility by introducing bidirectional transformation between classical DB and RDF data stores to enhance the sustainable capabilities of IoT systems for autonomous analytics. Two case studies, one on weather and another on heart-rate measurement collections through IoT sensors, are used to show the transformation process in operation.

Keywords: internet of things; autonomous analytics; data compatibility; resource description framework; semantic annotation

1. Introduction

Data science has become a major field of study, being recognized and taught internationally as a separate major. A major component of data science is data analytics. Now, when there is a discussion of data analytics, there are many ways, techniques, and tools to collect, analyze and test the available data in reforming it to obtain fruitful outcomes [1]. By contrast, data analytics plays a role in science to process raw data and draw highly sophisticated conclusions while extracting meaningful information from a burst of data. Data analytics helps in capturing useful patterns from raw data to be used for fast processing of results. Inferences and driving rules introduced in intelligence-based techniques and algorithms are best known for data [2,3]. These technologies and algorithms support industries and organizations in reducing research costs. The concerned fields for data analytics are Big Data, Internet of Things (IoT) [4], Machine Learning, Semantic Web (SW) and much more. When SW is used for data modeling, then Resource Description Framework (RDF), as a data model, plays a key role in ontology building, intelligent reasoning and data linking.

Furthermore, IoT at the point of data representation perspective uses RDF for smart objects while collecting data from sensors [4,5]. Data are collected from IoT sensors and then integrated through Semantic Web technologies to make them semantically readable from SPARQL queries [6]. IoT is an automatic environment in which data are being created, communicated, shared and utilized. The IoT environment can be used to integrate industries, societies, businesses and education to enjoy a better quality of life [7]. IoT is a collection of scattered smart objects and entities distributed with capabilities of sharing information with the use of sensors (as shown in Figure 1). These objects have embedded identification through radio-frequency identification (RFID) technology [4,8]. Despite its extensive spread development, IoT is still in its newborn phase and has considerable room for exploration in a variety of problems like standards, scalability, heterogeneity in different devices, common service description language, domain specific service discovery, and incorporation with current IT systems [9,10]. The foundation of IoT builds upon these RFID tags, information-sharing techniques, and sensors in addressing the abilities like tracking, visualizing and controlling of smart objects. It imparts the vision of the upcoming era concerning electrical and metallic objects being improved to be inter-linked and controlled to provide or overcome domain-specific facilities [11]. Such a phenomenon happens when transforming real-world objects into smart objects with the capability of sensing and communicating all around their compatible environment [2]. Technology like RFID is largely used for tracking real-life objects like people, animals, and electronic objects. RFID tags are encoded with Electronic Product Codes (EPCs), which are used for tracking objects in IoT [12]. Huge numbers of IoT devices from different vendors have dissimilar configuration and installation approaches. These devices have different methods for semantics and syntactic interoperability methods, so there will be semantic and syntactic conflicts. It will be difficult to add a new node in an IoT network without semantic ambiguity [13]. A lack of standards for interoperability without concerning the underlying hardware is a huge issue in IoT. Currently, IoT devices are domain specific and are used for special purposes [14].

Figure 1. IoT Network showing devices, sensors, and DB.

Information is presented using various data patterns and forms on the extensive scale. Emerging information with different data patterns and forms decreases device and database capacity to prepare for information investigation [15]. The dominant part of the framework takes a shot at conventional devices and databases. Information and preparing models need to focus on conquering similarity issues by adopting data transformation mechanisms [16]. The Semantic Web introduced RDF as a data model having the capability of capturing hierarchal relationships among different resources. This made a system to cover the gap of intelligence, which was long missing when using data for analysis and results [17]. Systems enhanced to handle data modeled in RDF are now able to capture reasoning with the help of available data. Data comprise multiple triples when combined to build-up a statement. A triple is a combination of a subject, a predicate, and an object. Also, the Web has been used as a common medium for applications to run regardless of Operating System and type of device. Furthermore, using W3C rules built in XML follows the structure of a web page. XML has the capability of introducing customized tags readable by a machine for further utilization in any application. XML is now used as a standard and common language for semantics when transforming data from one entity to another. This entity can be a device, an application, web-based sources and services and embedded software. XML data can be obtained from Relational DB, whereas Relational DB is a collection of relations (tables) built upon the relational data model. This study will introduce a Bidirectional Data Transformation Model (BDTM) for bi-directional data transformation by mapping of different data models and their customization, which brings them together using a common language. All data forms to a common language will reduce the loss of data during the transition of the transformation process [18]. The generic constructs belonging to each data model are devised to show BDTM-based transformation methodology of data in resolving update problems for the data found in both forms, i.e., RDB obtained from the IoT dataset and RDF.

This paper is organized into five sections. Section 2 is a literature review covering the evolutionary history of different data models along with tools and languages for transformation. In Section 3, the proposed model is described starting from the abstract level to a detailed-level representation along with various generic constructs derived from the data model after experimenting and observing similarities. Case studies based on IoT-based datasets are presented to see how this model helps in data transformation without considering about the loss, including results and discussion. Conclusion and References are presented in the last sections.

2. Literature Review

Nowadays, information is tending to ascend as fundamental in its significance for ventures, social orders, governments and financial matters for scientifically focused resolutions. Ordinary based data models can further be changed into semantic information models, particularly asset depiction systems (RDF) utilizing these specialists. The RDF in Semantic Web information model is utilized as a part of extensive information for data representation purposes [19]. Moreover, information from individual change instruments, devices, and models that originated from specific domain to utilize machine learning techniques. It performs with constant change for better usage and relevant results [1]. Accordance to the literature of data transformation, the next subsections are used to discuss the importance of data in the field of IoT involving a database, the evolution of a database, semantic annotation of data, querying XML and RDF, and transformation tools and techniques.

2.1. Internet of Things (IoT)

Interoperability permits the interfaces of a framework to work with another framework with no confined access or execution [20]. This interoperability can be syntactic, semantic or Cross-area from the viewpoint of associated web objects i.e., the Web of Things (WoT) [11]. Syntactic interoperability refers to intercommunication and information trade between two or more frameworks. Semantic interoperability explains deciphering the data traded seriously and precisely with a specific end goal to deliver characterized significant results to clients. By contrast, Cross-area interoperability covers various social, hierarchical, political, and legitimate substances that cooperate for a common interest and data trade [15].

Organization of these items set forth a significant rundown of vital, operational, strategic and mechanical difficulties particularly from the point of view of interoperability [6]. Interoperability is one of the greatest obstructions keeping organizations from receiving the IoT [4]. The absence of similar guidelines and calculations together increment the many-sided quality, wasteful aspects, client disappointment and the expense too. Displaying the knowledge of IoT articles can include a huge commitment in making this interoperability conceivable [20,21]. An inventive rundown of arrangements can be employed from computational insight areas (Fuzzy Logic, Neural Networks, Artificial Intelligence, Swarm Intelligence, and Genetic Algorithms), Machine adapting, Deep learning and their best in class augmentations. The fulfillment of such a commitment requires data representation to have relativity, correspondence, hierarchal relationship, and intelligence capabilities. There is a huge rush of information demanding proper mapping services for analyzing and visualization of data without loss in the wireless sensors networks [22,23], which further provide the help of RDB and RDF data models for representing IoT datasets [6].

2.2. Database for IoT Dataset

IoT-based datasets can be annotated to be capable of analyzing data smartly. Information collected from different sensors is summarized and plotted into a file or database for storage. The database can be in a normalized or non-normalized form. When we talk about the non-normalized form, then it is about data stored in NoSQL [24,25]. The normalized format is concerned mostly with a relational or tabular form of data storage. A NoSQL-based database management system (DBMS) known as MongoDB is used for data storage. By contrast, there are many relational database management systems like MySQL, Oracle, etc., which are used to store data in tabular format. The RDB data model has been introduced, whereas querying relational model-based data was first named as SEQUEL and was proposed by Chamberlain. SEQUEL was named as SQL and made a standard by the ANSI Committee in 1986. This evolution of history made data maintenance and management much more efficient and effective. IoT's sensor-based data collection can be linked and interpreted to analyze correspondence with other devices more easily with the use of RDBs.

2.3. Semantic Web for Data Representation

The RDF as an information model used in Semantic Web to determine hierarchal relationships among various assets. This framework used to cover the knowledge gap between utilizing information for examination and results [26]. It improved to handle information displayed in RDF that were developed with the assistance of accessible information. The information contained several triples when joined and developed in an announcement

2.4. Querying XML and RDF

Asset Description Framework (RDF) is utilized as a part of the Semantic Webs to allow machine reasoning of information among Webs. RDF Schemas (RDFS) help RDF characterize properties (qualities), sorts, and connections of assets in RDF records [26]. Questioning RDF/RDFS archives depends on tree traversal and basic example coordinating. From the profitability perspective, SQL asks for social databases, which are viewed as more straightforward and take less time to figure out than utilizing the RDF-based dialect, for example, SPARQL. Moreover, Social DBMS is maintained for the accessibility perspectives as well.

XSLT-based change is a good outline of syntactic change of XML records [27]. For our circumstance, XSLT (Standard for XSLT is available from the W3C Website https://www.w3.org/TR/xslt20/) is used for phonetic change between different XML-designs e.g. XPATH (Standard for XPath is available from the W3C Website https://www.w3.org/TR/xpath20/) Expressions are a possible course of action. Each report of a particular standard is changed into examination of an acknowledged structure in the midst of the phonetic change stage [16]. During the second stage, i.e., endorsed to-definitive semantic change, the acknowledged structure is changed into an integrated semantic standard structure, which is RDF for our condition.

2.5. Data Transformation Tools and Languages

Business Intelligence Software Tools, which are accessible, for the most part consider social databases. A change from RDF and RDFS to social databases with the goal that they are in the simple-to-use and broadly accessible structure is along these lines, i.e., a consistent methodology of information administration [1,28]. R2RML is a dialect for mappings from databases to RDF datasets, providing a separate language for relational data transformation into RDF data models, communicated in a structure and target vocabulary of the mapping creator's decision. This model could, for instance, offer an SPARQL endpoint mapped over the relation database model, or help in producing RDF store, or offer a Linked Data interface.

Endpoint can be mapped over the relational database model, or assist in producing RDF store, or offer a Linked Data interface.

This research is designed to compare and map currently available tools and technologies used for transformation between RDB and the Semantic Web [1]. As in Figure 2, tools are likely to be used for transformation between RDB to RDF known as D2R, D2RQ, R2O, Relational, OWL, Virtuoso RDF Views, DB2OWL, RDBtoOnto, Triplify, Ultrawrap, and R2RML. This study aims to reduce the response time of DB queries and offer compatibility with the semantically enriched data.

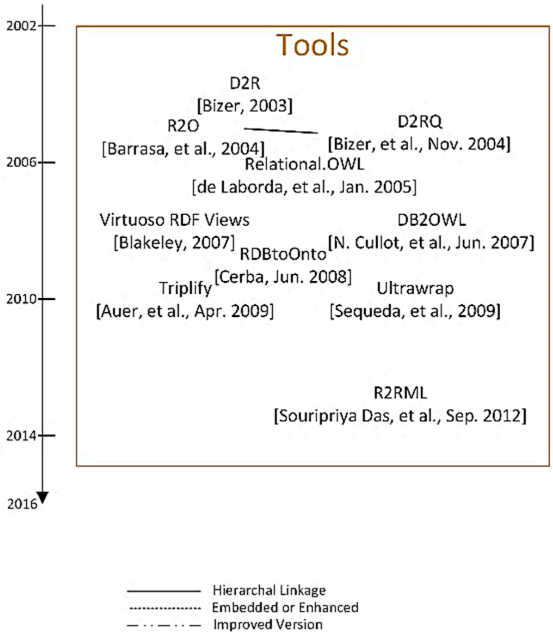


Figure 2. Transformation Tools Evolution.

3. Proposed Scheme: Bidirectional Data Transformation Model for IoT

The Semantic Web is a data model for intelligence due to its nature of linkage and unique identification of resources regarding data. It offers a huge opportunity for new concepts like IoT, to adopt latest data models for overcoming rushing in decision-based analytics using data linkage among resources. Many researchers that are working in IoT are syncing the Semantic Web data model as a part of IoT Architecture [21]. Syncing of different data models requires the modeling of a transformation mechanism having the capability of showing the fast impact of change appearing on any end of the data producer or receiver [16]. To make the system of IoT and semantic annotation more powerful, which works bidirectionally [29], two different data models need to have to a common language. XML is a common language, a popular language for better customization and distribution capabilities. This research is mostly concerned with data stored in a relational form collected from IoT infrastructure and then transformed into XML and RDF data formats. Due to this, IoT is able to semantically interpret while ignoring the underlying heterogenous hardware [30]. Data transformation is useful to XML to RDF databases collected from IoT devices.

Matching becomes easier and is possible when two different data structures are in an intermediate form based on the same format [7]. Data linking is done using RDF-based triplets representing a hierarchal relationship link between resources. Transformation is required for data and metadata to restore the data to their original form. This process can be done in a dramatic improvement in the enhanced utilization and semantics compatibility among systems. It can be used in incorporating large-scale loss democratic transformation of traditional systems into semantically enriched systems. A bidirectional transformation is lacking with respect to its full capacity [29]. The proposed methodology will directly provide benefit to systems, containing both traditional and semantic-enriched data.

Currently, available research tools and methods are used as state of the art for transformation. Bidirectional Transformation works properly to map different schema level transformations. In this research, we will look deeply into possible mappings, implementations, and updates and will test our results. Data are being collected from IoT devices and then the data are classified to apply mapping techniques. Mapping is done in a data transfer classifier section where tagging is used for semantic identification. These results will be analyzed and discussed based on the previous results. The structure of Class and Property for transformation to happen from the RDB Schema makes it possible for bidirectional transformation. The data format analysis is in a common language like XML; RDF storage and similar transformation back to RDB followed the XML data format (as shown in Figure 3).

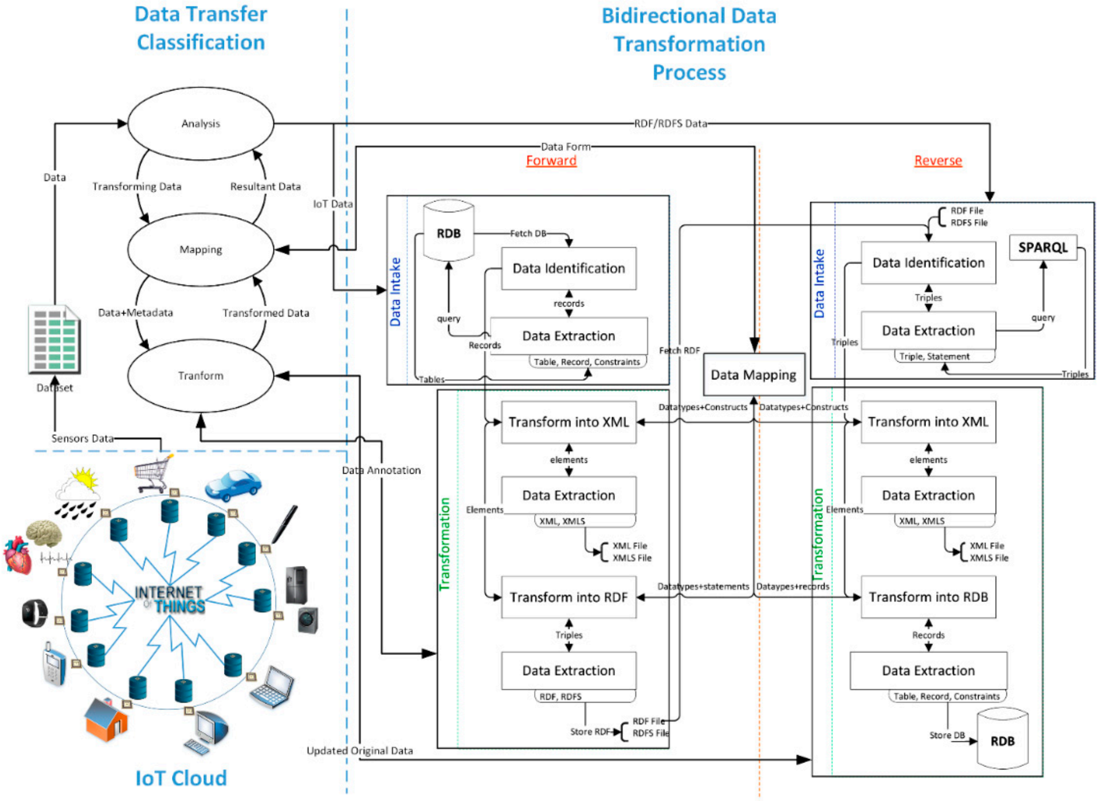
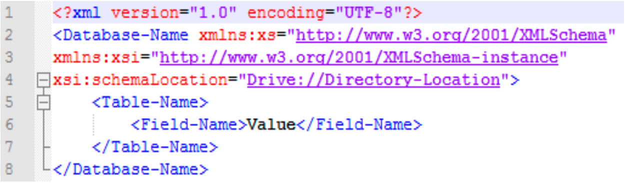


Figure 3. Complete Procedure of BDTM involving IoT.

Generic Constructs and Implementation

The information of a table's column is defined as a complex type while constraint information is defined as simple type element. A loop is started that selects names of a table one by one from the list. A query is executed to fetch the record set of the table (all tuples in this table) along with some basic information regarding column names and their data type. One more loop is designed to fetch information in the columns of the concerned table. If the value is an integer type, then it is type cast to store as a string. All information is also stored in ArrayList for future use to write to an XML file. This information is also forwarded to a list of the string type.

The header tag contains information regarding XML version. The schema tag provides information, starting with the name of the database. This tag also includes namespace information to show that elements and attributes originating from the source are identified. Local XML schema is also referred to in this tag, which helps validate the XML document against XSD. Location of the XSD file is mentioned as a resource in the XSI prefix. Introductory tags for XML data representation start and end with the database name. Values are stored under the tags of fields (as shown in Figure 4).

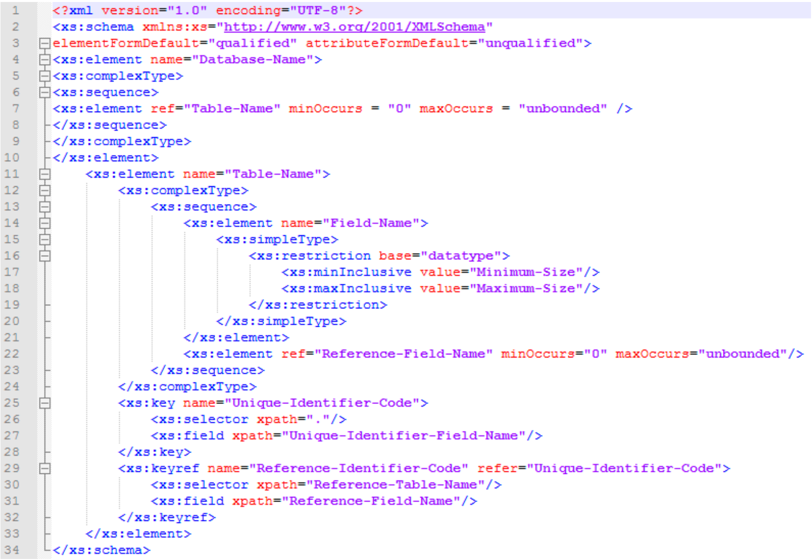


```xml
<?xml version="1.0" encoding="UTF-8"?>
<Database-Name xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="Drive://Directory-Location">
    <Table-Name>
        <Field-Name>Value</Field-Name>
    </Table-Name>
</Database-Name>
```

Figure 4. Generic XML Structure Defined for BDTM.

A class tag of RDF is created that contains a reference ID that is obtained from the XML element tag while reading the XSD file. It is encircled by tags, named as a label, comments, and subclass of tags. Another class tag is started that has a reference ID as the table name followed by the same tags. Before closing the class tag, a new tag is opened to write attributes/columns of the table. The subclass tag is used to define minimum cardinality restrictions along with data type of the attribute. For every attribute, two restriction tags are opened; one for minimum cardinality and second for maximum cardinality. Closing tag for the class is introduced. After that, properties for each attribute are registered and written. The sub tags used in property tag defined information like, sub property, label, comment, range and domain.

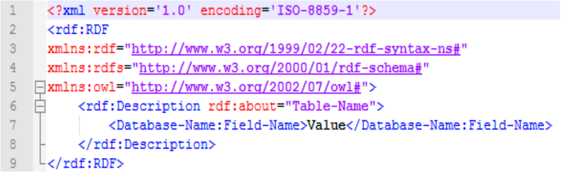
Names of tables with element tags are also recorded to obtain information about tables. The element tag has table information after DB information. Database information from columns is also captured along with their data types and minimum and maximum occurrences. Information of keys (primary and foreign keys) is also extracted from XSD. This process is repeated until XSD has acquired the information of all tables and their respective columns along with the keys. While reading the information of XSD, loops are introduced that are used to traverse XML schema. Of the inner loops, one is used for reading columns and constraints of a table while the second is used to find out keys in this table (as shown in Figure 5).



```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="Database-Name">
<xs:complexType>
<xs:sequence>
<xs:element ref="Table-Name" minOccurs = "0" maxOccurs = "unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element name="Table-Name">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Field-Name">
                    <xs:simpleType>
                        <xs:restriction base="datatype">
                            <xs:minInclusive value="Minimum-Size"/>
                            <xs:maxInclusive value="Maximum-Size"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element ref="Reference-Field-Name" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <xs:key name="Unique-Identifier-Code">
            <xs:selector xpath="."/>
            <xs:field xpath="Unique-Identifier-Field-Name"/>
        </xs:key>
        <xs:keyref name="Reference-Identifier-Code" refer="Unique-Identifier-Code">
            <xs:selector xpath="Reference-Table-Name"/>
            <xs:field xpath="Reference-Field-Name"/>
        </xs:keyref>
    </xs:element>
</xs:schema>
```

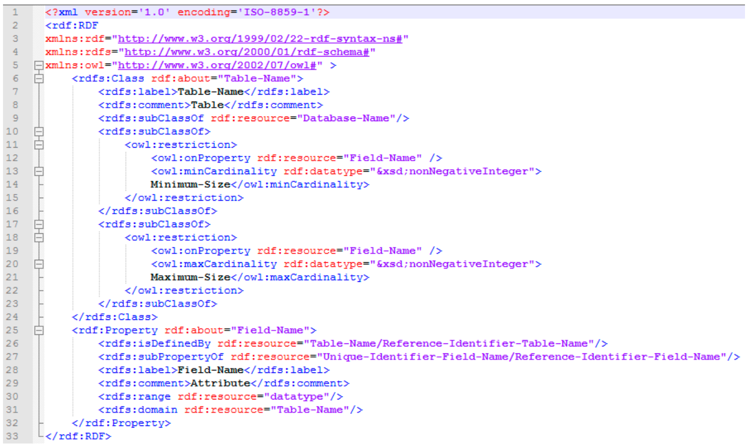
Figure 5. Generic XSD Structure Defined for BDTM.

In starting the process, introductory headers are included that contain information regarding the XML version. Second introductory tags are schema information labels, starting with the different namespace and URLs of RDF, OWL, and RDFS repository in W3C. RDF tags used for data representation open and close with the database name. Values are stored under the tag of fields obtained from the database name (as shown in Figure 6). Similarly, the schema for RDF is developed by taking care of constraints, cardinality, and field data types (as shown in Figure 7).



```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#">
    <rdf:Description rdf:about="Table-Name">
        <Database-Name:Field-Name>Value</Database-Name:Field-Name>
    </rdf:Description>
</rdf:RDF>
```

Figure 6. Generic RDF Structure Defined for BDTM.



```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" >
    <rdfs:Class rdf:about="Table-Name">
        <rdfs:label>Table-Name</rdfs:label>
        <rdfs:comment>Table</rdfs:comment>
        <rdfs:subClassOf rdf:resource="Database-Name"/>
        <rdfs:subClassOf>
            <owl:restriction>
                <owl:onProperty rdf:resource="Field-Name" />
                <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
                Minimum-Size</owl:minCardinality>
            </owl:restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:restriction>
                <owl:onProperty rdf:resource="Field-Name" />
                <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">
                Maximum-Size</owl:maxCardinality>
            </owl:restriction>
        </rdfs:subClassOf>
    </rdfs:Class>
    <rdf:Property rdf:about="Field-Name">
        <rdfs:isDefinedBy rdf:resource="Table-Name/Reference-Identifier-Table-Name"/>
        <rdfs:subPropertyOf rdf:resource="Unique-Identifier-Field-Name/Reference-Identifier-Field-Name"/>
        <rdfs:label>Field-Name</rdfs:label>
        <rdfs:comment>Attribute</rdfs:comment>
        <rdfs:range rdf:resource="datatype"/>
        <rdfs:domain rdf:resource="Table-Name"/>
    </rdf:Property>
</rdf:RDF>
```

Figure 7. Generic RDFS Structure Defined for BDTM Transformation of Weather and Cardiotocography Datasets.

Two cases are mentioned here pertaining to how annotation and transformation will look in different cases in IoT. One instance includes a dataset [31] on cardiotocography of 2126 measurements and classifications of fetal heart rate (FHR) signals. Another case includes the transformation process of data from a smart city weather dataset (CityPlus Weather Datasets http://iot.ee.surrey.ac.uk:8080/datasets.html) collected using IoT-based sensors [32]. Dataset representation as a graph is shown in Figures 8 and 9.

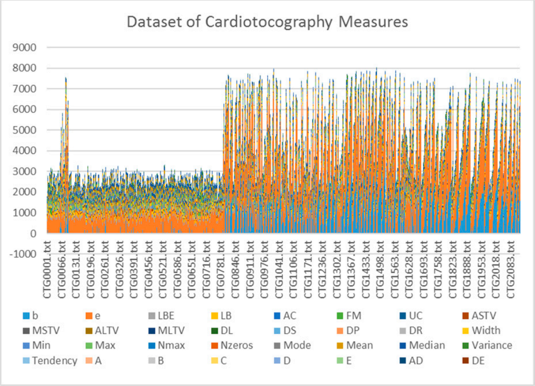


Figure 8. Graphical view of the Dataset of the Cardiotocographic Measures.

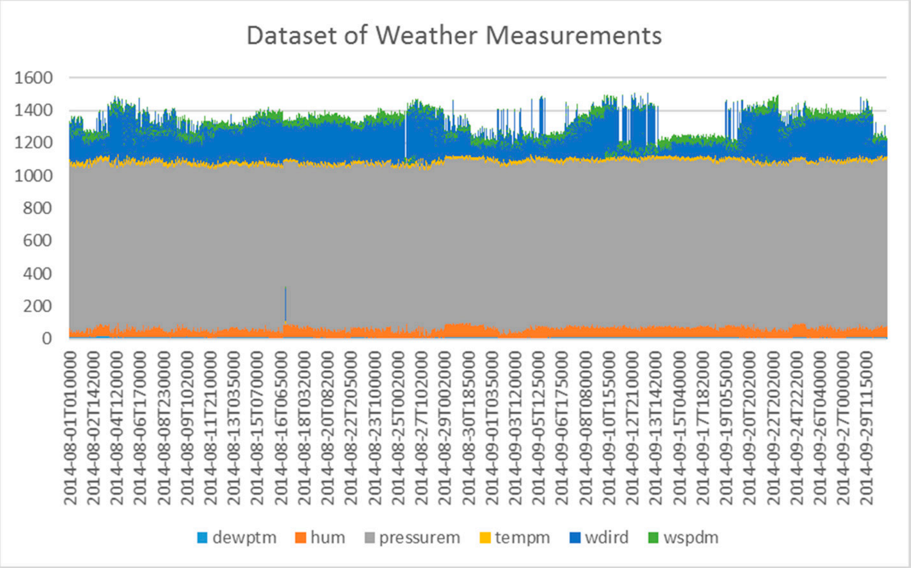


Figure 9. Graphical view of the Dataset of the Weather Measures.

4. Results and Discussion

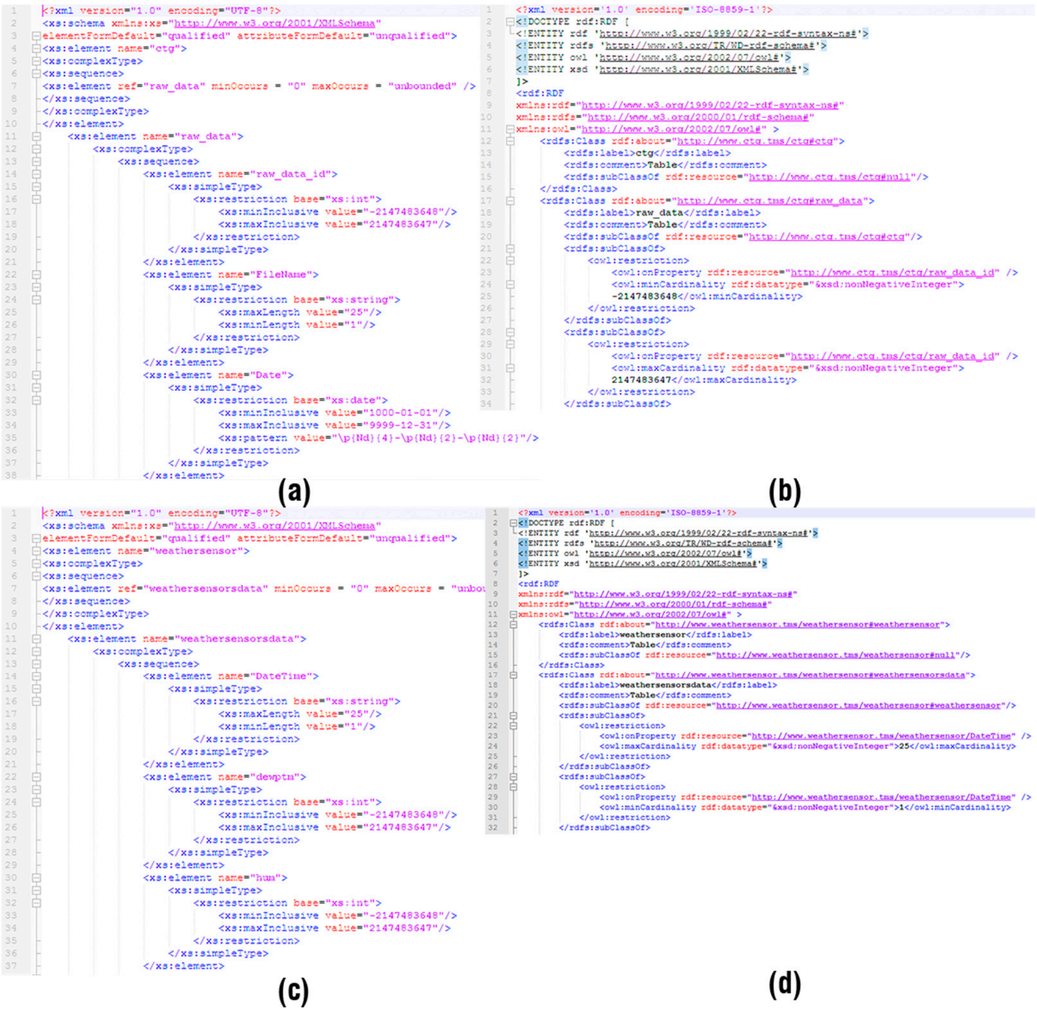
After execution of the transformation process from sensors, the data start with the collection of information regarding the names of all tables. The processes obtain information about the minimum and maximum occurrences along with primary key/composite key and foreign key constraints.

The information is forwarded to a list of texts while reading the basic information of the table; one more iteration is introduced to read the information of each field of the focused table. Column information consists of the name of the column and metadata with constraints. Another process reads information as minimum and maximum values of the current column along with data type. Two CTG and Weather datasets are used for bidirectional transformation to analyze our idea. The transformation is visible in a sample output taken from both datasets in metadata representation using the schema form of XML and RDF, as shown in Figure 10.

The delay factor in data retrieval and storage can be improved by the induction of bidirectional transformation among RDB and RDF schemas. Better results can be achieved with respect to data representation by mapping using a common languageOur idea can be enhanced during retrieval and transformation of data from either side, by mapping a common language. Birdirectional transformation is done to extract semantics from the data through RDF. Forward and backward mapping is required to follow the same form of the structure and data as shown in Figure 11. During RDF building the update problem is resolved by observing the time and resource significantly. Due to this factor, each change can result in a recreation of a whole RDF store from scratch.

In the Semantic Web, data are found in two forms, i.e., OWL (the majority) and RDF. Our research showed that OWL is closer to the intelligence domain specific problem, whereas RDF is generic. OWL is found in three different forms and can be considered as a rule oriented in the nature of data representation. Such representation can only be interpreted by a special system, which is truly built for Artificial Intelligence (AI). OWL is complex to make use of, even for other non-AI-based systems. Three different forms of OWL are OWL Lite, OWL DL, and OWL Full.

This opens IoT system capabilities to sustain information concerning a sensor's sources to obtain an analyzed form of data analytics. These analytical formations are based on rule-based intelligence using RDF or non-AI formats using NoSQL. These types of information are suitable for big data [33]. The RDF format is obtained through XML or RDB to support traditional indexing and analytics based on compositions using SQL queries.



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="ctg">
<xs:complexType>
<xs:sequence>
<xs:element ref="raw_data" minOccurs = "0" maxOccurs = "unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element name="raw_data">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="raw_data_id">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="-2147483648"/>
                            <xs:maxInclusive value="2147483647"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="FileName">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="25"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Date">
                    <xs:simpleType>
                        <xs:restriction base="xs:date">
                            <xs:minInclusive value="1000-01-01"/>
                            <xs:maxInclusive value="9999-12-31"/>
                            <xs:pattern value="\p{Nd}{4}-\p{Nd}{2}-\p{Nd}{2}"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
```

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<!ENTITY rdfs 'http://www.w3.org/TR/WD-rdf-schema#'>
<!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
<!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" >
    <rdfs:Class rdf:about="http://www.ctg.tms/ctg#ctg">
        <rdfs:label>ctg</rdfs:label>
        <rdfs:comment>Table</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.ctg.tms/ctg#null"/>
    </rdfs:Class>
    <rdfs:Class rdf:about="http://www.ctg.tms/ctg#raw_data">
        <rdfs:label>raw_data</rdfs:label>
        <rdfs:comment>Table</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.ctg.tms/ctg#ctg"/>
        <rdfs:subClassOf>
            <owl:restriction>
                <owl:onProperty rdf:resource="http://www.ctg.tms/ctg/raw_data_id" />
                <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
                -2147483648</owl:minCardinality>
            </owl:restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:restriction>
                <owl:onProperty rdf:resource="http://www.ctg.tms/ctg/raw_data_id" />
                <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">
                2147483647</owl:maxCardinality>
            </owl:restriction>
        </rdfs:subClassOf>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="weathersensor">
<xs:complexType>
<xs:sequence>
<xs:element ref="weathersensorsdata" minOccurs = "0" maxOccurs = "unbou
</xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element name="weathersensorsdata">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="DateTime">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="25"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="dewptm">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="-2147483648"/>
                            <xs:maxInclusive value="2147483647"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="hum">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="-2147483648"/>
                            <xs:maxInclusive value="2147483647"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
```

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<!ENTITY rdfs 'http://www.w3.org/TR/WD-rdf-schema#'>
<!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
<!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" >
    <rdfs:Class rdf:about="http://www.weathersensor.tms/weathersensor#weathersensor">
        <rdfs:label>weathersensor</rdfs:label>
        <rdfs:comment>Table</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.weathersensor.tms/weathersensor#null"/>
    </rdfs:Class>
    <rdfs:Class rdf:about="http://www.weathersensor.tms/weathersensor#weathersensorsdata">
        <rdfs:label>weathersensorsdata</rdfs:label>
        <rdfs:comment>Table</rdfs:comment>
        <rdfs:subClassOf rdf:resource="http://www.weathersensor.tms/weathersensor#weathersensor"/>
        <rdfs:subClassOf>
            <owl:restriction>
                <owl:onProperty rdf:resource="http://www.weathersensor.tms/weathersensor/DateTime" />
                <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">25</owl:maxCardinality>
            </owl:restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:restriction>
                <owl:onProperty rdf:resource="http://www.weathersensor.tms/weathersensor/DateTime" />
                <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
            </owl:restriction>
        </rdfs:subClassOf>
```

Figure 10. Output Sample: (**a**) XMLS Output for CTG dataset; (**b**) RDFS Output for CTG dataset; (**c**) XMLS Output for Weather dataset; (**d** RDFS Output for Weather dataset.

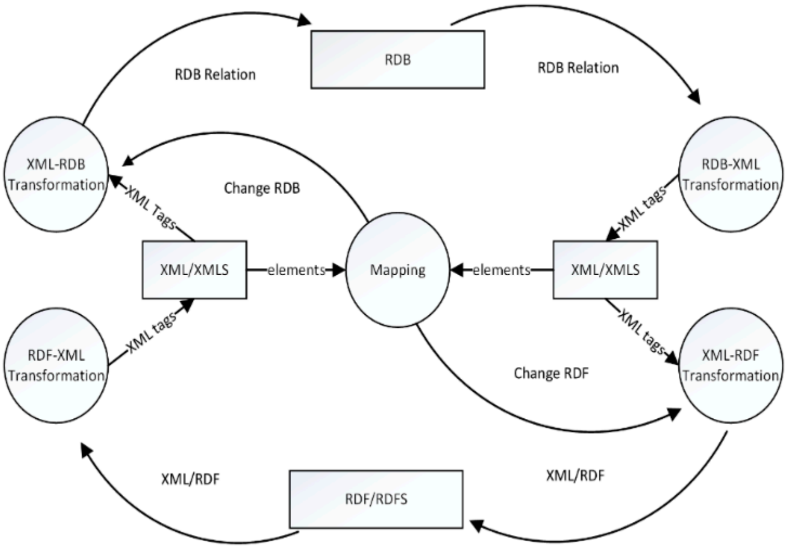


Figure 11. Mapping Playing a Key Role in Rectifying Change in BDTM.

5. Conclusions

Understanding the structure of an RDF data model is necessary for the smooth and target-oriented execution of the bi-directional data transformation process between IoT and Semantic Web environments. The RDF data model is built from class and property hierarchal relationships. Matching becomes easier and possible when two different data structures are transformed into a common intermediate form. IoT devices sync with semantic data models to extract data, and then mapping techniques are applied for tagging. The information tables are formed in which all the processes with minimum and maximum occurrences along with primary/foreign key constraints occur. This information is extracted from data in the form of XM and RDF. Then, bidirectional transformation occurs from XML to RDF to make it semantically interoperable among IoT devices. This experiment showed that OWL is intelligence domain specific while RDF is generic and can be used in any domain. This study will benefit standard structure conversions and data transformations of a simple sensor's oriented dataset into ontologies/RDF triples. This comes with lossless improvement through an effective methodology of data transformation. This methodology provides data compatibility and classification representation using semantic annotation for the IoT-based dataset. Although much work is already being done in this regard, a single solution that can provide a single smart operation has not been found. This study is hoped to become a benchmark in semantic annotation concerning the IoT environment. This involves enhancing data compatibility to be sustainable for autonomous analytics. Our methods of converting the IoT dataset into XML and then into RDF do not depend on the database; therefore, any database with known data types can be managed using the proposed system to obtain standard conversions devised by the World Wide Web Consortium (W3C).

Author Contributions: Kaleem Razzaq Malik and Masood Habib conceived and designed the experiments. Kaleem Razzaq Malik and Farhan Ullah performed the experiment. Shehzad Khalid, Muhammad Umar and Awais Ahmad analyzed the data. Taimur Sajjad contributed reagents/materials/analysis tools. Farhan Ullah and Taimur Sajjad contributed in the introduction, and literature review of paper. Kaleem Razzaq Malik, Shehzad Khalid and Awais Ahmad contributed in Proposed Model and Analysis. Muhammad Umar and Masood Habib contributed in Results and Discussion section of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Steenbeek, J.; Coll, M.; Gurney, L.; Mélin, F.; Hoepffner, N.; Buszowski, J.; Christensen, V. Bridging the gap between ecosystem modeling tools and geographic information systems: Driving a food web model with external spatial-temporal data. *Ecol. Model.* **2013**, *263*, 139–151. [CrossRef]
2. Yang, K.; Cho, S.B. Towards sustainable smart homes by a hierarchical hybrid architecture of an intelligent agent. *Sustainability* **2016**, *8*, 1020. [CrossRef]
3. Farhan, M.; Aslam, M.; Jabbar, S.; Khalid, S. Multimedia based qualitative assessment methodology in elearning: Student teacher engagement analysis. *Multimed. Tool. Appl.* **2016**. [CrossRef]
4. Malik, K.R.; Ahmad, T.; Farhan, M.; Ullah, F.; Amjad, K.; Khalid, S. Multiagent semantical annotation enhancement model for iot-based energy-aware data. *Int. J. Distrib. Sens. Netw.* **2016**. [CrossRef]
5. Jabbar, S.; Minhas, A.A.; Imran, M.; Khalid, S.; Saleem, K. Energy efficient strategy for throughput improvement in wireless sensor networks. *Sensors* **2015**, *15*, 2473–2495. [CrossRef] [PubMed]
6. Jara, A.J.; Olivieri, A.C.; Bocchi, Y.; Jung, M.; Kastner, W.; Skarmeta, A.F. Semantic web of things: An analysis of the application semantics for the iot moving towards the iot convergence. *Int. J. Web Grid Serv.* **2014**, *10*, 244–272. [CrossRef]
7. Khan, M.; Din, S.; Jabbar, S.; Gohar, M.; Ghayvat, H.; Mukhopadhyay, S. Context-aware low power intelligent smarthome based on the internet of things. *Comput. Elect. Eng.* **2016**, *52*, 208–222. [CrossRef]

8. Shen, J.; Tan, H.; Moh, S.; Chung, I.; Wang, J. An efficient RFID authentication protocol providing strong privacy and security. *J. Int. Tech.* **2016**, *17*, 443–455.
9. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243.
10. Mao, J.; Zhou, Q.; Sarmiento, M.; Chen, J.; Wang, P.; Jonsson, F.; Xu, L.; Zheng, L.; Zou, Z. A hybrid reader tranceiver design for industrial internet of things. *J. Ind. Inform. Integr.* **2016**, *2*, 19–29. [CrossRef]
11. Zhang, X.; Zhang, J.; Li, L.; Zhang, Y.; Yang, G. Monitoring citrus soil moisture and nutrients using an iot based system. *Sensors* **2017**, *17*, 447. [CrossRef] [PubMed]
12. Xiao, G.; Guo, J.; Da Xu, L.; Gong, Z. User interoperability with heterogeneous iot devices through transformation. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1486–1496. [CrossRef]
13. Liu, F.; Tan, C.-W.; Lim, E.T.; Choi, B. Traversing knowledge networks: An algorithmic historiography of extant literature on the internet of things (iot). *J. Manag. Anal.* **2017**, *4*, 3–34. [CrossRef]
14. Chen, Y. Industrial information integration—A literature review 2006–2015. *J. Ind. Inform. Integr.* **2016**, *2*, 30–64. [CrossRef]
15. Park, J.H.; Hung, J.C.; Yen, N.Y.; Jeong, Y.S. Guest editorial: Advanced convergence technologies: Big data, iot, cloud computing. *J. Int. Technol.* **2014**, *15*, 589–592. [CrossRef]
16. Cunha, J.; Fernandes, J.P.; Mendes, J.; Pacheco, H.; Saraiva, J. Bidirectional transformation of model-driven spreadsheets. In Proceedings of the International Conference on Theory and Practice of Model Transformations, Prague, Czech Republic, 28–29 May 2012; Springer: Berlin, Heidelberg; pp. 105–120.
17. Lee, C. Guest editorial: Automated big data analysis for social multimedia network environments. *Multimed. Tool. Appl.* **2016**, *75*, 12663–12667. [CrossRef]
18. Duan, L.; Binbasioglu, M. An ensemble framework for community detection. *J. Ind. Inform. Integr.* **2017**, *5*, 1–5. [CrossRef]
19. Malik, K.R.; Ahmad, T.; Farhan, M.; Aslam, M.; Jabbar, S.; Khalid, S.; Kim, M. Big-data: Transformation from heterogeneous data to semantically-enriched simplified data. *Multimed. Tool. Appl.* **2016**, *75*, 1–21. [CrossRef]
20. Jabbar, S.; Ullah, F.; Khalid, S.; Khan, M.; Han, K. Semantic interoperability in heterogeneous iot infrastructure for healthcare. *Wirel. Commun. Mob. Comput.* **2017**. [CrossRef]
21. Huang, Y.; Li, G. A semantic analysis for internet of things. In Proceedings of the 2010 International Conference on Intelligent Computation Technology and Automation (icicta), Changsha, China, 11–12 May 2010; IEEE: Piscataway, NJ, USA; pp. 336–339.
22. Hammoudeh, M.; Newman, R.; Dennett, C.; Mount, S.; Aldabbas, O. Map as a service: A framework for visualising and maximising information return from multi-modalwireless sensor networks. *Sensors* **2015**, *15*, 22970–23003. [CrossRef] [PubMed]
23. Abuarqoub, A.; Hammoudeh, M.; Adebisi, B.; Jabbar, S.; Bounceur, A.; Al-Bashar, H. Dynamic clustering and management of mobile wireless sensor networks. *Comput. Netw.* **2017**, *117*, 62–75. [CrossRef]
24. Reissmann, S.; Frisch, D.; Pape, C.; Rieger, S. Correlation and consolidation of distributed logging data in enterprise clouds. *Int. J. Adv. Int. Technol.* **2014**, *7*, 39–51.
25. Trzaska, M. Easy development of web applications using webodra2 and a dedicated ide. *Int. J. Adv. Int. Technol.* **2013**, *6*, 156–169.
26. Cyganiak, R.; Wood, D.; Lanthaler, M. Rdf 1.1 concepts and abstract syntax. *W3C Recomm.* **2014**, *25*, 1–8.
27. Kay, M. Xsl Transformations (xslt) Version 2.0 w3c Recommendation. Available online: https://www.w3.org/TR/rdf11-concepts/ (accessed on 17 April 2017).
28. Das, S.; Sundara, S.; Cyganiak, R. R2rml: Rdb to Rdf Mapping Language. Available online: https://www.w3.org/TR/r2rml/ (accessed on 17 April 2017).
29. Koffel, E.; Kroenke, K.; Bair, M.J.; Leverty, D.; Polusny, M.A.; Krebs, E.E. The bidirectional relationship between sleep complaints and pain: Analysis of data from a randomized trial. *Heal. Psychol.* **2016**, *35*, 41–49. [CrossRef] [PubMed]
30. Gil, D.; Ferrández, A.; Mora-Mora, H.; Peral, J. Internet of things: A review of surveys based on context aware intelligent services. *Sensors* **2016**, *16*, 1069. [CrossRef] [PubMed]
31. Bache, K.; Lichman, M. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml/ (accessed on 17 April 2017).

32. Ali, M.I.; Gao, F.; Mileo, A. Citybench: A configurable benchmark to evaluate rsp engines using smart city datasets. In Proceedings of the International Semantic Web Conference, Bethlehem, PA, USA, 11–15 October 2015; Springer: Cham, Switzerland; pp. 374–389.
33. Ochian, A.; Suciu, G.; Fratu, O.; Suciu, V. Big data search for environmental telemetry. In Proceedings of the 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Odessa, Ukraine, 27–30 May 2014; IEEE: Piscataway, NJ, USA; pp. 182–184.