Journal of Theoretical and Applied Information Technology
15th September 2024. Vol.102. No. 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# DATA INTEGRATION APPROACHES AND DATA CLASSIFICATION ALGORITHMS: A REVIEW

**MOHD KAMIR YUSOF[1], WAN MOHD AMIR FAZAMIN WAN HAMZAH[2], SUHAILAN SAFEI[3]**

[1,2,3] Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut, Terengganu, Malaysia

E-mail: [1]anis@gmail.com, [2]kamir2020@gmail.com, amirfazamin@unisza.edu.my
[4]suhailan@unisza.edu.my

**ABSTRACT**

This article is focuses on study the different data types in data integration and the most suitable approach for data integration to support the different data types. Three types of data; structured data, semi-structured and unstructured data will be used in data integration for specified purposes such as data analysis, etc. However, the challenge in data integration is to provide a unified view or standard view and improve the accuracy during data retrieving process. The purpose of this paper is to review current approaches in data integration and classification algorithms for classification of the data. Four (4) current approaches in data integration which are federation, mediator, mashup, extract-load-transform (ETL). Meanwhile, four (4) classification algorithm which are support vector machine (SVM), naive bayes network, decision trees, and neural network. Based on findings, mediator approach produced better performance in term of response time, memory usage, and CPU usage compared to federation, mashups, and ETL. Meanwhile, SVM algorithm indicates more accurate for data classification compared to naive bayes network, decision trees, and neural network. Finally, based on these findings, a new model in data integration has been proposed to overcome the issues related with unified view or standard view and accuracy in data classification.

**Keywords:** *Database, Data integration, Structured Data, Semi-Structured Data, Unstructured Data, Classification Data*

## 1. INTRODUCTION

In this modern era, databases have been commonly used by organizations to store the data and retrieve the data according to their business nature [1]. Nowadays, data is coming from different data sources. Common data types in most of data sources is structured [2], semi-structured [3] and unstructured data [4]. These types of data will be used for analyzing purposes and help a decision maker to decide in their business organization. However, the challenges in data analyzing from different data sources and different data types are data integration by provides a unified view and data classification for accuracy purposes. A unified view or standard view is needs to allow different applications or systems access the data [5]. Meanwhile, accuracy is important during data retrieving process from different data sources for data classification. This paper is focuses to find out the best approach in data integration and the best algorithm in data classification. Section 2 describes about database, and data types will be explained in section 3. Section 4 explained about data integration and Section 5 discussed about data classification. Findings about data integration, data classification and a proposed model for data integration will be discussed in Section 6. Finally, conclusion will be described in section 7.

## 2. DATABASE

A database is a structured collection of consistent and structured records or data that is stored in a computer so that it can be consulted by a program to answer queries [6-10]. Database system is a computer-based to record and maintain information/data. The purpose of database is to store information to allow organizations to produce report, mailing labels, inventory, etc. An organization must have accurate and reliable data for effective decisions making. The organizations are required to maintain records on the various facets maintaining relationship among them. There are several advantages of database; easy to retrieve information quickly and flexibly, easy to store large quantities of

information, easy to organize and reorganize information, and easy to print and distribute data in a variety of ways. However, data in database can be in different format of data such structured, semi-structured and unstructured data. Database must be able to handle and manage different format of data especially during extraction process.

## 3. DATA TYPE

Data type can be divided into three (3) categories. There are structured data, semi-structured data, and unstructured data.

### 3.1 Structured Data

It may be distinct as a statistics model and is usually to be had in text layout simplest. Easy to locate. It's miles in relational databases and factories. Generally, it will be generated by means of special packages which includes human or machine, airline reserving machine, stock control, ERP device and CRM gadget. Examples of based records are date, telephone, quantity, social security quantity, credit score card number, client call, cope with, product name and variety, transaction facts as well [2]. Fully structured data consists of defined data kinds that have styles that facilitate search. This shows information types with an excessive corporation level, like statistics in relational databases. Established records is information whose factors may be triumph over for effective evaluation. It's been organized into a database, which is a structured archive. This is true for any data that can be stored in rows and columns in a square database [11]. They have contact keys and can be simply mapped into predefined fields. Today, this type of data is the most popular and simplest way to store records in development.

#### 3.1.1 Relational database

Relational database is one of the approach has been used to store data in one or more tables of columns and rows. Usually, every table has key attributes which are the primary and foreign key. It can be used to differentiate one line from some other line and is capable of explaining the entity's characteristics. Through understanding the existence of primary key and foreign key, it can analyze the relationship among the entities. Figure below shows the analysis result of identify the primary and foreign keys [12]:

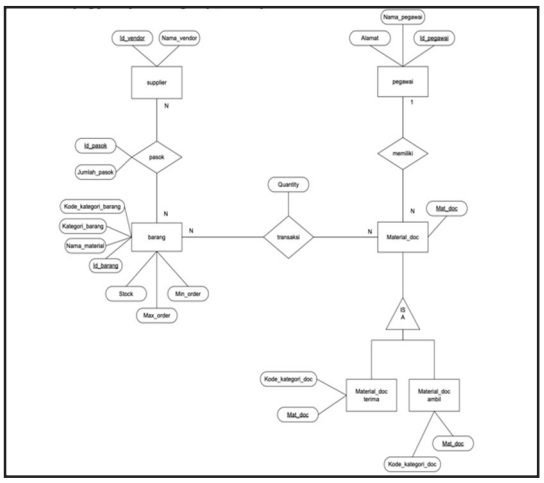



*Figure. 3 Analysis Result of Identity the Primary and Foreign Key [12]*

There are specific types of databases which approaches to use them and the regions of the software are analyzed. A relational database used to store and analyze specific data from behavioral trials. Three distinctive signs are analyzed, respectively, reminiscence required to keep the data, time to load the records from an external record into pc memory and generation time throughout all statistics through one cycle [13]. Traditional relational databases' inability to accommodate complicated structures and user-defined record types is overcome by object-based/XML databases, making the conversion of object-oriented databases, object-relational databases, and XML are all hot research topics [14]. Studies societies use relational databases that are available for free to mine new data for their research projects. These databases are liable to security problems. The reliability of the data source has to be authenticated earlier than the use of it for any software motive. As a consequence, to know the owner and reliability of data, watermarking is carried out to the data. Digital Watermarking programs for Relational Database. To preserve relational databases RDB, diverse watermarking techniques have been carried out for achieving numerous goals like a database [15].

#### 3.1.2 PostgreSQL

PostgreSQL is known as an open source relational database control device. PostgreSQL has organization-magnificence capabilities which includes SQL windowing features, the capability to create combination features and also utilize them in window constructs, not unusual table and recursive not unusual table expressions, and streaming replication. Those functions are hardly ever

discovered in different open source databases but are common in more recent variations of proprietary databases along with Oracle, SQL Server, and DB2. PostgreSQL permits you to put in writing stored techniques and features in numerous programming languages. similarly to the prepackaged languages, you may allow help for extra languages through the usage of extensions. You may even write combination features in any of those languages, thereby combining the data-aggregation power of SQL with the native abilities of every language to gain extra features than you can with the language on your own. You can also write functions in C and lead them to callable, similar to some other stored feature. it could even define mixture features containing nothing but SQL. Each custom type supported in PostgreSQL is state-of-the-art and very easy to apply, rivaling and frequently outperforming most different relational databases [16].

Except for the polygon intersection queries, the interpretation is based on real-world business scenarios and the following queries, which, in addition to their basic infrastructures, corroborate the ubiquity of PostgreSQL in almost all circumstances. The PostgreSQL database management system is a free and open-source object-relational database management system. The SQL language is used to communicate with the PostgreSQL database. BTree, Generalized Inverted Indexes (GIN), Hash, and Generalized Search Tree (GiST) (R-tree-over-GiST) are all examples of generalized inverted indexes are just a few of the index types supported by PostgreSQL. TBTree is the default index type, which works with all data types and can be used to successfully perform equality and variety queries. GiST indexes are used by PostgreSQL to index data in several forms of geometric statistics as well as full-text searches, for preferred balanced tree systems and excessive-space spatial queries [17]. The PostgreSQL database era, it supports the great majority of SQL transactions, concurrent management, and modern functionalities, it is the world's most advanced open-source DBMS. such as complex queries, triggers, perspectives, transactional integrity, and the ability to add data type extensions, features, operators, and procedural languages to the database. Table partitioning, which allows for faster querying of certain data, can help improve performance in these situations. This method decreases the number of physical reads at the database while queries are being processed. It is possible to speed up the mission while sorting [18].

### 3.1.3 SQLite

Although SQLite is recognized for its limited development capabilities, a device that can assist in the development and use of SQLite databases might be extremely useful for inexperienced Android developers. One of the alternatives is SQLite, a light-weight relational database that comes pre-installed with Android OS. SQLite is a popular relational database that supports SQL syntax, transactions, and well-structured statements. SQLite is a popular database engine when compared to other database engines. Special data types are supported by SQLite, as well as other databases, including text, INTEGER, and real, to mention a few. The primary class for SQLite databases is SQLite Database. This course will show you how to open and close database connections, as well as conduct CRUD operations. execSQL() is a technique for simultaneously running multiple SQL statement [19]. It's a robust graphical user interface that connects to a relational database management system. A personal SQLite server is used to process and store all the information. While inventory levels may be lower than more secure levels, the application provides all current day stock control capabilities for product In-Out, income data, and additional signals. The structured query language in Python requires the use of SQLite. Python and structured query language are used to make this possible, SQLite queries should be employed in accordance with the requirements as shown in Figure 4 [20].

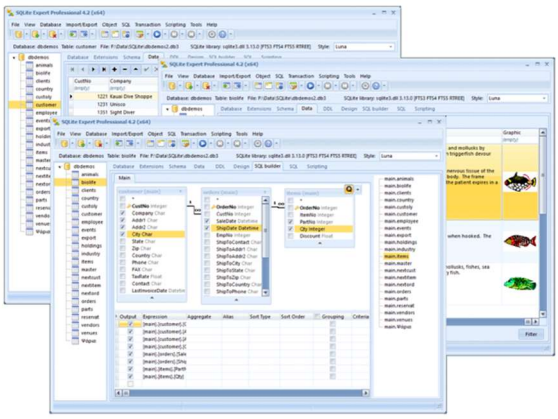


*Figure 4: SQLite design*

SQLite is a lightweight database management system that is self-contained, requires no configuration, and operates independently (no longer as a consumer/server). It's found in a wide range of applications, including web browsers, mobile devices, and embedded systems. SQLite is built on the foundation of the operating system,

which is the interfaces for digital memory, file management, and tool drivers are included in this package [21]. CreateFile(), deleteFile(), resizeFile(), flushFile(), and flushAll() are the five functions in the improved file API (). These are the commands that are used to communicate with the SQLite database.

SQLite is a method library that implements a self-contained, server-less, and configuration-free transactional square database engine. The code for SQLite is open source, which means it can be used for both business and personal uses. SQLite is the world's most used database, with a large list of applications and several high-profile projects. SQLite is a free and open source embedded relational database. It was first released in 2000 with the goal of providing a convenient mechanism for applications to control information without the overhead associated with dedicated relational database control systems. SQLite is known for being extremely portable, simple to use, compact, efficient, and dependable. SQLite is a relational database management system embedded in a C programming library, according to several academics. SQLite is not a consumer-server database engine when compared to many other database control models. Alternatively, it might be incorporated into the final programmed [22].

### 3.1.4 MongoDB

MongoDB is a file-based NoSQL datastore and has been used in the industrial world. Despite being non-relational, many relational database functions are implemented in MongoDB, including advanced querying techniques include sorting, secondary indexing, variety queries, and nested file searching. Manual indexing, indexing on embedded files, and indexing location-based records are all supported, as are actions like create, insert, read, update, and remove. Data is stored in document collections, which are entities that give managed data structure and encoding. Figure 5 shows the replication of MongoDB. In this architecture, files are stored using the BSON binary encoding and serialized as JavaScript object Notation (JSON) objects internally. There are no schema regulations in MongoDB, therefore it may be useful for multi-attribute lookups on data and semi-structure data with unique key-value pairs [23]. XML, JSON, YALM, and CSV are examples of semi-structured documents.

MongoDB is a report-oriented database with a replication architecture that allows customer applications to determine the consistency and latency trade-offs they want to make on a per-operation basis. One of the fundamental purposes of the MongoDB replication machine is to provide surprisingly accessible scattered information storage, allowing customers to directly choose the bulk of the trade-offs possible in a replicated database machine that are not required in single node structures. MongoDB's method for changeable consistency was inspired by knowledge of disaster frequency and how it interacts with the consistency [24].

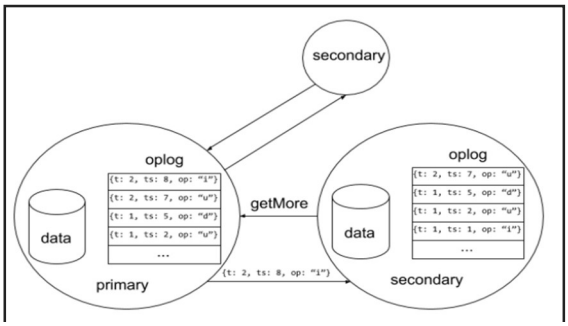



*Figure. 5 Replication Architecture of MongoDB [25]*

It included causal consistency in version 3.6, which allows clients to toggle on or off an extra set of consistency guarantees. MongoDB is a document-oriented database that stores records using JSON-like devices. All facts in MongoDB are stored in JSON or BSON binary format. In a MongoDB database, a fixed collection is a group of files that are all the same. MongoDB uses the WiredTiger storage engine, which is a transactional key-value statistics store that manages the connection to a local, long-lasting garage media. MongoDB also lets you run a database as a reproduction set, which is a group of MongoDB nodes that functions as a consensus group and maintains a logical duplicate of the database state on each node. MongoDB replica units rely on frontrunners and use a consensus procedure akin to the Raft technique [25].

### 3.2 Semi-Structured Data

Semi structured data forms consist of built -in tags and marks that divide data values, enabling clustering and hierarchy of information. Databases and documents can consist of semi -structured data types. This type of data only covers about 5% to 10% of structured, semi structured, or unstructured ocean data, but it has significant commercial use [3]. It's a

structure that doesn't follow the same formal structural data models as relational databases or other data table formats but incorporates tags or other markers to separate the semantic elements and perform the essential hierarchical record and data field tasks [26]. Semi structured data is data that isn't stored in a database but has a few organizational qualities that make it easier to examine. With few processes, we can keep in the relational databases even sometimes it can be a little bit hard for semi structured data, but it can make the space.

### 3.2.1 XML

Extensible Markup Language (XML) is one of the data formats that is supported by Data Reading. It loads the collection of data from the disk into memory to be processed. All of them are accommodated through a simple attribute-value pairs model. The URI prefix string must be identified when choosing the data format, XML especially if the data format is originally structured. Figure 6 shows sample of XML format. The namespace of the dataset, table, and credential also must be prepared to record the output in a database or endpoint [27].

```
<menu>
 <area text="Welcome" file="index.html">
  <submenuitem text="New in Scribus 1.5" file="readme.html"/>
  <submenuitem text="Specification" file="specs.html"/>
 </area>
 <area text="Documentation" file="intro.html">
  <submenuitem text="Introduction" file="documentation.html">
   <submenuitem text="Editorial Notes" file="editorial.html"/>
   <submenuitem text="About the Team" file="about1.html"/>
  </submenuitem>
  <submenuitem text="Setup" file="config.html">
   <submenuitem text="Configuring Scribus" file="settings1.html"/>
   <submenuitem text="Hyphenation and Spellchecking" file="hyphenator.html"/>
   <submenuitem text="Font Setup" file="fonts1.html"/>
   <submenuitem text="Fonts in Depth" file="fonts2.html"/>
  </submenuitem>
  <submenuitem text="Scribus Basics" file="about2.html">
   <submenuitem text="Quick Start Guide" file="qsg.html"/>
   <submenuitem text="Command Line Reference" file="cli.html"/>
   <submenuitem text="Keyboard Shortcuts" file="keys.html"/>
   <submenuitem text="Mouse Shortcuts" file="mouse.html"/>
   <submenuitem text="Document Information" file="docinfo.html"/>
   <submenuitem text="Working with Frames" file="WwFrames.html"/>
   <submenuitem text="Working with Text" file="WwText.html"/>
   <submenuitem text="Text Properties" file="TextProp.html"/>
   <submenuitem text="Search and Replace" file="SearchReplace.html"/>
   <submenuitem text="Working with Styles" file="WwStyles.html"/>
   <submenuitem text="Working with Images" file="WwImages.html"/>
```

*Figure. 6 XML data*

XML format is an acceptable method that records and sorts semi-structured data in Character Large Objects (CLOB) because different data types can be recorded in different XML nodes. XML language is analyzed and mostly used in algorithms for mining semi-structured data. XML files are also used in conversion methods for web semi-structured data [28].

### 3.2.2 JSON

JavaScript Object Notation (JSON) is in demand day by day. The data act as arbitrarily complex hierarchies and have a human-readable plain text format. Developers admit that JSON is one of the semi-structured data formats that provides more flexibility for their design. Database systems analyze a lot of JSON data and JSON objects will be recorded as a string or binary of the objects will be used [29]. Popular social media like Facebook and Twitter which contain JSON APIs are the example of big public JSON data sets where the data is collected, and the proprietary data may be enriched when the system is logged in. Figure 7 shows sample of JSON format.

```
{
    "orders": [
        {
            "orderno": "748745375",
            "date": "June 30, 2088 1:54:23 AM",
            "trackingno": "TN0039291",
            "custid": "11045",
            "customer": [
                {
                    "custid": "11045",
                    "fname": "Sue",
                    "lname": "Hatfield",
                    "address": "1409 Silver Street",
                    "city": "Ashland",
                    "state": "NE",
                    "zip": "68003"
                }
            ]
        }
    ]
}
```

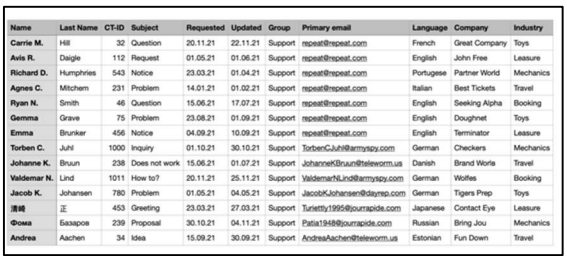
*Figure. 7 JSON data*

JSON has a small size, processes quickly, can summarize the data structure presentation, and is not loaded with a lot of data details. Simply, JSON format can be written and read faster where a standard JavaScript function can parse it same as XML format. JSON can present semi-structured data by supporting their necessary hierarchical structure, record data in text document, and interpret the data of an individual and in device. Necessary data can be fetched in documents, any source of data formats and languages can be controlled, and objects that emerge by the infrastructure production can be adjusted and implemented permanently by this JSON technology [30].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 3.2.3 CSV

Today, Comma-Separated Values (CSV) is one of the data formats that is commonly used to share data publicly and the companies also publish tabular data on the web. As a semi-structured data, CSV files are exploited after the proposal of mapping languages where it has some features to handle the tabular data efficiently. For instance, a statistic that contains y-axis and x-axis as a Figure 8 [31].



| Name | Last Name | CT-ID | Subject | Requested | Updated | Group | Primary email | Language | Company | Industry |
|---|---|---|---|---|---|---|---|---|---|---|
| Carrie M. | Hill | 32 | Question | 20.11.21 | 22.11.21 | Support | repeat@repeat.com | French | Great Company | Toys |
| Avis R. | Daigle | 112 | Request | 01.05.21 | 01.06.21 | Support | repeat@repeat.com | English | John Free | Leasure |
| Richard D. | Humphries | 543 | Notice | 23.03.21 | 01.04.21 | Support | repeat@repeat.com | Portugese | Partner World | Mechanics |
| Agnes C. | Mitchem | 231 | Problem | 14.01.21 | 01.02.21 | Support | repeat@repeat.com | Italian | Best Tickets | Travel |
| Ryan N. | Smith | 46 | Question | 15.06.21 | 17.07.21 | Support | repeat@repeat.com | English | Seeking Alpha | Booking |
| Gemma | Grave | 75 | Problem | 23.08.21 | 01.09.21 | Support | repeat@repeat.com | English | Doughnet | Toys |
| Emma | Brunker | 456 | Notice | 04.09.21 | 10.09.21 | Support | repeat@repeat.com | English | Terminator | Leasure |
| Torben C. | Juhl | 1000 | Inquiry | 01.10.21 | 30.10.21 | Support | TorbenCJuhl@armyspy.com | German | Checkers | Mechanics |
| Johanne K. | Bruun | 238 | Does not work | 15.06.21 | 01.07.21 | Support | JohanneKBruun@teleworm.us | Danish | Brand Worls | Travel |
| Valdemar N. | Lind | 1011 | How to? | 20.11.21 | 25.11.21 | Support | ValdemarhLind@armyspy.com | German | Wolfes | Booking |
| Jacob K. | Johansen | 780 | Problem | 01.05.21 | 04.05.21 | Support | JacobKJohansen@dayrep.com | German | Tigers Prep | Toys |
| 清純 | 正 | 453 | Greeting | 23.03.21 | 27.03.21 | Support | Turletty1995@jourrapide.com | Japanese | Contact Eye | Leasure |
| Фома | Бахаров | 239 | Proposal | 30.10.21 | 04.11.21 | Support | Patia1948@jourrapide.com | Russian | Bring Jou | Mechanics |
| Andrea | Aachen | 34 | Idea | 15.09.21 | 30.09.21 | Support | AndreaAachen@teleworm.us | Estonian | Fun Down | Travel |

*Figure. 8 Example of CSV [36]*

There is a speculative approach that usually succeeds for statistical and syntactic settings of the CSV format in parallelizing parsing systematically in a distributed environment. Syntax error can be discovered in CSV data if the speculation-based approach is powerful. A complete chunk and logical CSV file record must be validated so all the data and record can be connected and speculated successfully [32].

### 3.2.4 HTML

Hypertext Markup Language (HTML) is in demand for its characteristics particular to a single web page. To retrieve and collect data, every web page needs training data [33]. HTML template extracts object strings and produces detail pages by inserting data from the database. HTML tags that create columns and rows are limited when used for object strings, predicates, and subjects which will be inspected and recognized. Lately, web tables act as a source of facts that can retrieve and collect data are advantageous for question, and answer systems [34].

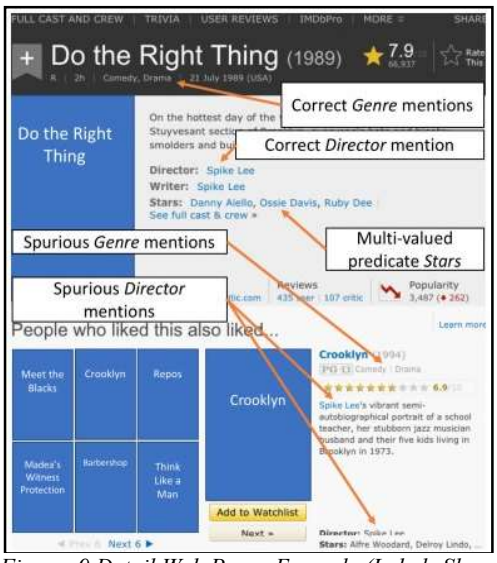



*Figure. 9 Detail Web Pages Example (Labels Show Annotation Challenges) [35]*

Normally, Web Pages have a lot of data. Web Pages can possibly have a great amount of use cases, text fields, and entities that can match with other entities. Extraction and annotation time will be more challenging, and more expenses need to be spent if more entities and the relation between them wanted to be examined. However, many entities in a Web Pages can cause something false and reduce the extraction process. Thus, the data will be less accurate [35]. We can visit hypertext documents or Web Pages either in the same or different site because Web Pages data have hyperlinks which means the pages are connected but we may encounter unwanted information. As we know, Web Pages do not have a fixed layout where the pages are rich in information such as texts, images, ads, etc. Web Pages that have no control over the information because we can upload and update any type of content on the pages. For example, books for sales websites that we can upload and update the book name, description, price, etc. [36].

### 3.2.5 NoSQL

Not Only SQL (NoSQL) is an essential alternative to relational databases and particularly referred to as non-relational databases. NoSQL databases can be classified into document, key value, and wide column stores and graph databases. These four classes also can be differentiated based on the foundation of functional and non-functional attributes. The functional attributes involve aggregation, atomicity, de-normalization, joins, and keys. Then, the non-functional attributes involve complexity, flexibility, performance, scalability, and

structure. NoSQL is well-known as having high efficiency and storage. Thus, NoSQL databases can store data uniformly, operate at a given time, and deal with faults [37]. Table 1 and Table 2 shows functional and non-functional attributes of NoSQL.

*Table 1: Functional Attributes to Differentiate NoSQL Classification [37]*

| S. No. | Features | Key Value Store | Document Store | Wide Column store | Graph Store |
|---|---|---|---|---|---|
| 1. | **Denormalization** | Applicable | Not Applicable | Applicable | Applicable |
| 2. | **Single Aggregate** | Applicable | Applicable | Applicable | Not Applicable |
| 3. | **Atomicity** | Applicable | Applicable | Applicable | Not Applicable |
| 4. | **Unordered Keys** | Applicable | Not Applicable | Not Applicable | Not Applicable |
| 5. | **Derived Table** | Not Applicable | Not Applicable | Applicable | Not Applicable |
| 6. | **Composite Key** | Not Applicable | Not Applicable | Applicable | Not Applicable |
| 7. | **Composite Aggregation** | Applicable( ordered) | Not Applicable | Applicable | Not Applicable |
| 8. | **Aggregation** | Applicable | Applicable | Applicable | Not Applicable |
| 9. | **Aggregation and Group by** | Applicable | Applicable | Not Applicable | Not Applicable |
| 10. | **Adjacency Lists** | Applicable | Applicable | Not Applicable | Not Applicable |
| 11. | **Nested Sets** | Applicable | Applicable | Not Applicable | Not Applicable |
| 12. | **Joins** | Not Applicable | Not Applicable | Not Applicable | Not Applicable |

*Table 2: Non-functional attributes to differentiate NoSQL classification [37]*

| Data model | Performance of queries | Scalability of data | Flexibility of schema | Structure of database | Complexity of values |
|---|---|---|---|---|---|
| **Key-value store** | High | High | High | Primary key with some value | None |
| **Column Store** | High | High | Moderate | row consisting multiple columns | Low |
| **Document Store** | High | Variable (High) | High | JSON in form of tree | Low |
| **Graph Database** | Variable | Variable | High | Graph – entities and relation | High |

### 3.3 Unstructured Data

Unstructured data is not like a predefined data model. It can be in text, image, sound, video, or other formats [4]. It's hard to find. That is found in applications, SQL databases, Data Lakes, and Data Warehouse. Humans and machines can both build word processing software, presentation software, email clients, and media viewing and editing tools. It can be found in a variety of places, including text files, reports, email communications, audio files, video files, pictures, and surveillance imagery [38]. Usually, unstructured data is not easily searchable, like in video, audio, and social media posts [39]. The available data do not fit neatly in the relational database [40]. Because unstructured data is not organized in a prescribed manner or has an established data model, it is incompatible with traditional relational databases. Businesses are using unstructured data in a variety of business intelligence and analytics applications as it becomes increasingly widespread in IT systems. As a result, there are alternative platforms to store and handle this data.

#### 3.3.1 Text

Text can be located as text documents. Most of the standard enterprise files consist of word processing documents, notes, PDFs and presentations are unstructured records. text additionally can be discovered as text mining that is an understanding - in depth method that represents the evaluation by means of files. Essentially it is delivered for descriptive purposes, and it has fast advanced by way of including methods capable of classifying files according to their latent topic or to infer approximately the "sentiment" of clients or the customers of social networks. The improvement to these techniques has gone together with the evolution of both the computational performance of the algorithms necessary to analyze textual data and the technology needed to keep data. textual content typically determined in marketing campaigns, emblem control, fraud detection and other lawsuits. It also facilitates claims management and repayment, subrogation, relationships between the assist center and clients and also can analyze contracts' clauses. The table beneath shows the numerical representation of files [41].

| | Terms | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Are | Autonomous | Car | Cars | Claims | Change | Driving | Insurance | Look | Motor | No | Self |
| **Document 1** | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| **Document 2** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

*Figure. 10 Represent Numerical of Documents [41]*

#### 3.3.2 Email

Email represents a precious source of data that may be harvested for knowledge, reengineering and repurposing undocumented commercial enterprise strategies of businesses and establishments. emails are grouped in keeping with the procedure model they belong to. This is observed through sub-grouping and labeling the emails of every technique model into commercial enterprise activity types.

Email is the most used communication medium, and is considered through some because it is the first and biggest social medium. While email's initial purpose was to replace (private) letters between people, it is now utilized for a variety of functions ranging from event planning to file sharing and modification to managing the execution of multi-person complicated tasks. Due to its extensive use in personal, however most significantly, expert contexts, email represents a precious source of data that may be harvested for knowledge, reengineering and repurposing undocumented enterprise techniques of corporations and institutions [42]. Email may be considered as unstructured data and also semi-structured data, email messages may be in text fields that aren't usually effortlessly analyzed. There may be a framework that can collect data on email sender behaviors depending on the distribution of global sending. Any IP address that sends an email message will be compared and given a confidence value. Spammers no longer send spam from a single IP address; instead, they use many IP addresses to send spam [43].

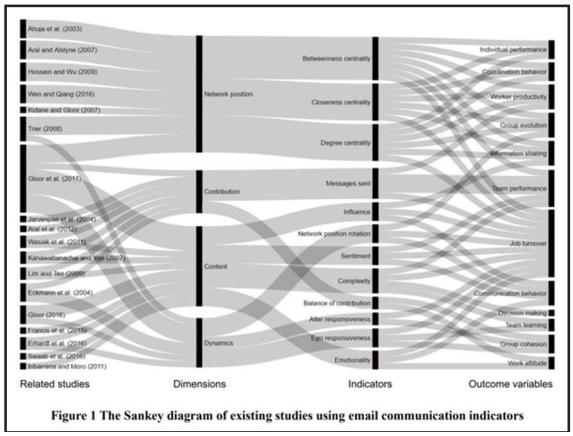



*Figure. 11 Existing studies employing email communication indicators as depicted in the Sanky diagram [44]*

## 4. DATA INTEGRATION

Data integration is combining data residing at different data sources and providing the user with a unified view of these data [45- 46]. The purpose of data integration is to integrate data from different data sources. Figure 12 shows, each data source contain different type of data which is structured data, semi-structured data, or unstructured.

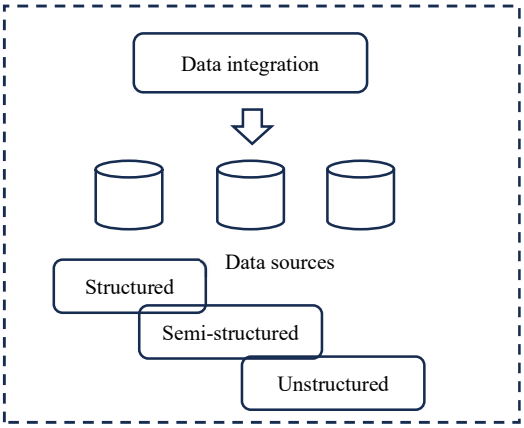



*Figure. 12: Data integration from different data sources*

Three different characteristics are used in data integration which are architecture, mapping approach and data format. Four (4) approaches are currently implemented in data integration which are federation, mediator, Extract Transform Load (ETL), and mashups.

### 4.1. Federation

Federation is referring to an architecture in which middleware, consisting of a relational database management system and provides uniform access to several heterogeneous data sources [47 - 48]. A database federation is like single database management system which is users can search for information and manipulate data using the full power of the SQL language. In this approach, a single query may access data from multiple sources, joining and restricting, aggregating, and analyzing the data.

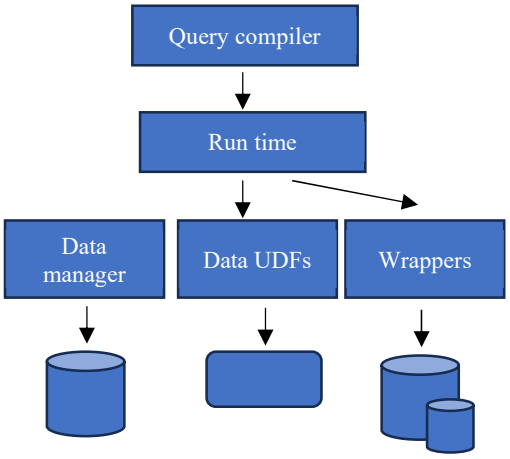



Fig. 13: Federation architecture

Database federation is employing a database engine to create a virtual database from several data sources. A database engine is a key driver, but the method by which data or functions are included in the federation differs. Federation is powerful approach, but it is not a panacea. Federation integrates data lazily, as it is needed [49]. This method is suitable for medium size of data sets or when queries are selective enough that only a small fraction of the data ever be returned. Federation is only work well, when not much execution of pre-processing or cleansing.

### 4.2. Mediator

A mediator is important component in data integration to supports a mediation schema and the distributed sources [46]. Mediation systems can be classified according to the approach used to define the mappings between data sources, global schema, and local schema [50 - 51]. A mediator approach has two main components: mediator and wrapper which is represented in Figure 2.6. A middle-layer services provides by mediator as an information integrator between the application and wrappers [52]. Mediator is responsible for retrieving information from data sources, transforming received data into a common representation, and integrating the homogenized data. Meanwhile, tasks of wrapper are composed of the schema translation processor (STP), query translation processor (QTP) and data translation processor (DTP). A responsibility of STP is to translate the data definition of objects in data sources from each source definition to Mediated Data Definition Language (MDDLs). QTP is responsible for translating a sub-query to specific query which is depends on the type of the query language used in each data source. DTP is responsible for transforming schemas to common data model.

### 4.3. Extract Transform Load (ETL)

ETL is a process of extract, transform and load [53 - 54]. ETL is not considered as a one-time event as new data added into Data Warehouse (DW) periodically in monthly, daily or hourly basis [55]. Three processes of ETL such as extraction, transformation and load are represented in Figure 2.8. The complexity of data extraction is usually due to flat files data source formats which have no structured relationship. Process of transformation is to make some cleaning and conforming of incoming data to gain accurate data, which is correct, consistent, and unambiguous. In this process, the extracted data are cleansed and transformed based on the business requirements. The last process of ETL is load. Load process is responsible to load the data processed by the first two processes into DW. Two methods have been applied in load process are refreshing and updating. The refreshing method is used to load the data into database during process of creating a DW, and the updating method is used to maintain the DW.

### 4.4. Mashup

Mashups is one of data integration approach which is combining content from multiple services or sources at run time [56 - 57]. Mashups are typically integration of data from heterogeneous data sources such as databases, search engines, or local files. Three important components in mashups approach are wrapper, script, and transformer. Wrappers are used to transform the data into a source-specific and self- describing XML structure for uniform data access within the framework. Result of XML structured is described as a set of objects, such as people, products, publications, or addresses. A set of scripts are used to handle queries from web GUI and web service. Querying data sources or web services is an integral part of data integration. In this operation, queries are directed towards a specific source and can be defined either explicitly or implicitly. Explicit queries are directed towards a specific source by the user interface via a web browser. Implicit queries are applied for XML document and define a query strategy on the input document.

## 5. DATA CLASSIFICATION

After data integration is done, the next process is data retrieval. One of the methods can be used in data retrieval is classification of supervised learning algorithms [58]. The purpose of classification is o forecast group membership for data instances from data source [59]. A few algorithms in classification which are support vector machine (SVM), naive bayes, decision tree, and neural network.

### 5.1. Support Vector Machine (SVM)

Support Vector Machine (SVM) models is related to classical multilayer perceptron neural networks [60]. In this model, a hyperplane is divided into two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances. Based on the result, it is proven to reduce an upper bound on the expected generalization error [61].

### 5.2. Naive Bayes (NB) Networks

Bayesian networks is composed by directed acyclic graphs with only one parent (representing the unobserved node) and several children (corresponding to observed nodes). By using this approach, assumption of independence among child nodes become strong in the context of their parent [62]. Measurement of independence model (Naive Bayes) is calculated based on estimating [63]. Based on previous result, Bayes classifiers is less accurate compared to others sophisticated learning algorithms (such as ANNs). However, performed a large-scale comparison of the naive Bayes classifier with state-of-the-art algorithms for decision tree induction, instance-based learning, and rule induction on standard benchmark datasets, and found it to be sometimes superior to the other learning schemes, even on datasets with substantial feature dependencies [64].

### 5.3. Decision Trees

Decision Trees (DT) are trees by sorting them based feature values to classify the instances. In this approach, each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume [65]. Decision tree is used a predictive model which maps observations about an item to conclusions about the item's target value in data mining and machine learning. More descriptive names for such tree models are classification trees or regression trees [66]. Post-pruning technique has been used in decision tree to evaluate the performance by using validation set. Any node can be removed and assigned the most common class of the training instances that are sorted to it [65].

### 5.4. Neural Network

The main advantages of Neural Networks (NN) are performing several regressions and/or classification tasks at once, compared to other network is only one. In most cases, therefore, the network will have a single output variable, although in the case of many-state classification problems, this may correspond to a few output units (the post-processing stage takes care of the mapping from output units to output variables). ANN is defined by the current values of weight to fixed problem related to activation and network architecture. ANN is defined by the current values of the weights. The weights of the net to be trained are initially set to random values, and then instances of the training set are repeatedly exposed to the net. The values for the input of an instance are placed on the input units and the output of the net is compared with the desired output for this instance. Then, all the weights in the net are adjusted slightly in the direction that would bring the output values of the net closer to the values for the desired output. There are several algorithms with which a network can be trained [67].

## 6. FINDINGS AND FUTURE WORK

### 6.1. Data integration approaches

A few applications have been developed by using different approaches in data integration as presented in Table 4. Three different parameters which are response time (ms), memory usage (%), and CPU usage (%) has been used to measure the performance among the applications. Mediator approach is the most popular approach has been used which is 13 applications compared to other mashup (1), ETL (1), and federation (1) in Table 3. Two different data model or data sources has been extracted which are structured data and semi-structured data. In extraction process, Resource Description Framework (RDF) /extensible markup language (XML)/ Global as View (GAV) mapping are used to extract and load the data. However, according to the results in Table 3, mapping approach using XML as mapping to extract and load semi-structured data produce better performance compared to RDF and GAV as a mapping approach. Three indicator measurement level has been used to measure the

performance which are low, medium, and high. In term of response time, if the result indicates low, processing time to execute query is slow. If the result indicates medium, processing time to execute the query is medium between low and high. If the result indicates fast, processing time to execute the query is fast. The best performance in term of response time is high because of fast query execution time. In term of memory usage and CPU usage, if the result indicates low, memory usage and CPU usage has been used to process the query is low. If the result indicates medium, memory usage and CPU usage has been used to process the query is medium. If the result indicates high, memory usage and CPU usage has been used to process the query is high. The best performance in term of memory usage and CPU usage is low because of less memory usage and CPU has been used to execute the query. According to the result in Table 3, mediator approach for data integration is more powerful in term of response time, memory usage and CPU usage compared to mashup, federation, and ETL.

### 6.2. Data classification algorithms

Four (4) different algorithms have been compared using smaller dataset (384) and large dataset (768). Table 4 and Table 5 show the performance of the algorithms. SVM algorithm is consistently produce better performance in term of correctness which is 72.92% for smaller dataset and 77.34% for large data compared to other algorithms. Performance of SVM algorithm is slightly increase by 1% - 5% average compared to other algorithms [81].

*Table 3: Comparison of data integration approaches*

| System | Approach | Data models | Mapping | Response time | Memory | CPU |
|---|---|---|---|---|---|---|
| LSM [68] | Mashup | Structured | RDB, XML | Medium | High | High |
| Arens [69] | Mediator | Structured | RDB, GAV | Medium | Medium | Medium |
| Garlic [70] | Mediator | Structured | RDB | Medium | High | Medium |
| InfoMaster [71] | Mediator | Structured | RDB | Medium | Medium | High |
| KRAFT [72] | Mediator | Structured | RDB | Medium | Medium | Medium |
| Observer [73] | Mediator | Structured | RDB | Medium | Medium | High |
| TSIMMIS [11] | Mediator | | RDB | | | |
| dbXML [74] | Mediator | Semi-structured | XML | Medium | Medium | Medium |
| Xindice [75] | Mediator | Semi-structured | XML | Medium | High | High |
| BioWarehouse [76] | Mediator | Semi-structured | RDB | Medium | Medium | Medium |
| OrientX [77] | ETL | Structured | XML | Medium | Medium | Medium |
| MIOMIS [78] | Mediator | Semi-structured | RDB, XML | Medium | Medium | Medium |
| FuhSen [79] | Federation | Structured | RDF, XML | Medium | Medium | Medium |
| Barkeley [80] | Mediator | Semi-structured | XML | High | Low | Low |
| eXist [80] | Mediator | Semi-structured | XML | High | Low | Low |
| Sedna [80] | Mediator | Semi-structured | XML | High | Low | Low |

*Table 4: Comparative of classification algorithms (smaller dataset)*

| Algorithm | Time (sec) | Correctly Classified | Incorrectly classified |
|---|---|---|---|
| SVM | 0.04 | 72.92% | 27.08% |
| Random Forest | 0.42 | 71.88% | 28.13% |
| Naïve Bayes | 0.01 | 70.57% | 29.43% |
| Decision Tree | 0.03 | 64% | 36% |
| Neural Network | 0.17 | 59% | 41% |

*Table 5: Comparative of classification algorithms (large dataset)*

| Algorithm | Time (sec) | Correctly Classified | Incorrectly classified |
|---|---|---|---|
| SVM | 0.09 | 77.34% | 22.66% |
| Random Forest | 0.03 | 76.30% | 23.70% |
| Naïve Bayes | 0.81 | 75.13% | 24.87% |
| Decision Tree | 0.14 | 73.83% | 26.17% |
| Neural Network | 0.23 | 72.40% | 27.60% |

Based on findings data classification algorithms by using two different datasets (small and large), SVM is produce better result compared to random forest, native bayes, decision type, and neural network.

### 6.3. Future Work

Based on finding in data integration approaches, and data classification algorithm, a new model will be proposed to integrate data from different data sources. Figure 14 shows the proposed model for data integration. A mediator approach will be used to extract the data from different data sources. Then, the data will be converted into semi-structured (XML). SVM algorithm will be used to classify the data. Based on this model, process of data classification is hopefully in data integration from different data sources become more efficient in term of response time and accurate in term of accuracy.

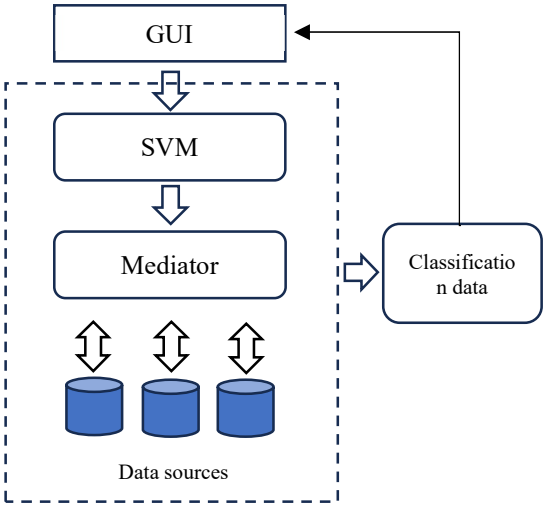



*Figure 14: A new model for data integration*

## 7. CONCLUSION

Data integration is important for organization to analyze the data from different sources to decide the decision in their organizations. A suitable approach in data integration is needs to integrate the sources. Then, a suitable algorithm also is needs to improve the accuracy in data classification. In this paper, a new model for data integration has been proposed based on findings in data integration approaches and data classification algorithms. This proposed model is hopefully can solve the problems related to data integration is producing a unified view of standard view and improve the accuracy during retrieving process.

## ACKNOWLEDGEMENT

This research was supported by Ministry of Higher Education (MOHE) through Fundamental Grant Scheme (FRGS/1/2020/ICT06/UNISZA/02/3).

## REFERENCES

[1] G. Li, X. Zhou, J. Sun, X. Yu., Y. Han, L. Jin, S. Li, "An autonomous database system", *Proceedings of the VLDB Endowment*, Vol 14, No. 12, 2021, pp. 3028-3042.

[2] K. H. Tan, & Y. Zhan, "Improving new product development using big data: a case study of an electronics company", *R&D Management*, Vol. 47, No. 4, 2017, pp. 570-582.

[3] M. S. Dickson, & P. O. Asagba, "The Semi-Structured Data Model and Implementation Issues for Semi-Structured Data", *International Journal of Innovation and Sustainability*, No. 3, 2017, pp. 47-51.

[4] A. A. Hussien, "Comparison of machine learning algorithms to classify web pages", *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 11, 2017.

[5] E. L. Sibanda, K. Webb, C. A. Fahey, M. S. K. Dufour, S. I. McCoy, C. Watadzaushe, F. M. Cowan, "Use of data from various sources to evaluate and improve the prevention of mother-to-child transmission of HIV programme in Zimbabwe: a data integration exercise", *Journal of the International AIDS Society*, Vol. 23, 2020, e25524.

[6] J. Berrington, "Databases. Anaesthesia & Intensive Care Medicine", Vol. 15, No. 2, 2014, pp. 56 - 61.

[7] E. Turban, C. Pollard, "Wood, G. Information Technology for Management: Advancing Sustainable, Profitable Business Growth, 10th Edition", *International Student Version (10th ed.): Wiley*, April 2015.

[8] R. A. Elmasri, & S. B. Navathe, "Fundamentals of Database Systems (5th Edition)", *Addison-Wesley Longman Publishing Co*., Inc, 2006.

[9] K. Berg, T. J. Seymour, R. Goel, "History of Databases", *International Journal of Management and Information Systems*, Vol. 17, No. 1, 2012, pp: 29 – 36.

[10] S. Bouamama, "Migration from a Relational Database to NoSQL", *International Journal of Knowledge-Based Organizations*, Vol. 8, No. 3, 2018.

[11] G. Li, "Human-in-the-loop data integration", *Proceedings of the VLDB Endowment*, Vol. 10, No. 12, 2017.

[12] J. H. Lubis, & E. M. Zamzami, "Relational database reconstruction from SQL to Entity Relational Diagrams", *In Journal of Physics: Conference Series*, Vol. 1566, No. 1, 2020, p. 012072, IOP Publishing.

[13] R. Kraleva, V. Kralev, & N. Sinyagina, "Design and Analysis of a Relational Database for Behavioral Experiments Data Processing", *International Journal of Online Engineering*, Vol. 14, No. 2, 2018.

[14] A. Maatuk, A. Ali, & N. Rossiter, "A framework for relational database migration", 2019.

[15] A. Alqassab, & M. Alanezi, "Relational Database Watermarking Techniques: A Survey", *In Journal of Physics: Conference Series*, Vol. 1818, No. 1, 2021, p. 012185). IOP Publishing.

[16] R. O. Obe, & L. S. Hsu, "PostgreSQL: Up and Running: a Practical Guide to the Advanced Open Source Database", *O'Reilly Media, Inc*, 2017.

[17] A. Makris, K. Tserpes, G. Spiliopoulos, D. Zissis, & D. Anagnostopoulos, "MongoDB Vs PostgreSQL: A comparative study on performance aspects. GeoInformatica", No. 25, 2021, pp. 243-268.

[18] A. Viloria, G. Acuna, D. J. A. Franco, H. H. Palma, J. P. Fuentes, & E. P. Rambal, "Integration of data mining techniques to PostgreSQL database manager system", *Procedia Computer Science*, No. 155, 2019, pp. 575-580.

[19] I. Musleh, S. Zain, M. Nawahdah, & N. Salleh, "Automatic Generation of Android SQLite Database Components", *In SoMeT*, 2018, pp. 3-16

[20] K. Yuvaraj, G. M. Oorappan, K. K. Megavarthini, M. C. Pravin, R. Adharsh, & M. A. Kumaran, "Design and Development of An Application for Database Maintenance in Inventory Management System Using Tkinter and Sqlite Platform", *In IOP Conference Series: Materials Science and Engineering*, Vol. 995, No. 1, 2020, p. 012012). IOP Publishing.

[21] W. Thompson, R. Karne, A. Wijesinha, & H. Chang, "Interoperable SQLite for a bare PC", *In International Conference: Beyond Databases, Architectures and Structures*, 2017, pp. 177-188). Springer, Cham.

[22] M. J. A. Ghali, & S. S. A. Naser, "ITS for Data Manipulation Language (DML) Commands Using SQLite", 2019.

[23] A. Makris, K. Tserpes, G. Spiliopoulos, & D. Anagnostopoulos, "Performance Evaluation of MongoDB and PostgreSQL for Spatio-temporal Data", *In EDBT/ICDT Workshops*, 2019.

[24] W. Schultz, T. Avitabile, & A. Cabral, "Tunable consistency in mongodb", *Proceedings of the VLDB Endowment*, Vol. 12, No. 12, 2019, 2071-2081.

[25] A. M. Dissanayaka, R. R. Shetty, S. Kothari, S. Mengel, L. Gittner, & R. Vadapalli, "A review of MongoDB and singularity container security in regard to hipaa regulations", *In Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, 2017, pp. 91-97

[26] Y. Lin, Z. Jun, M. Hongyan, Z. Zhongwe, & F. Zhanfang, "A method of extracting the semi-structured data implication rules", *Procedia computer science*, 131, 2018, pp. 706-716.

[27] G. Papadakis, L. Tsekouras, E. Thanos, G. Giannakopoulos, T. Palpanas, & M. Koubarakis, "The return of jedai: End-to-end entity resolution for structured and semi-structured data", *Proceedings of the VLDB Endowment*, Vol. 11, No. 12, 2018, pp. 1950-1953.

[28] Y. Lin, Z. Jun, M. Hongyan, Z. Zhongwei, & F. Zhanfang, "A method of extracting the semi-structured data implication rules", *Procedia computer science*, 131, 2018, pp. 706-716.

[29] W. M. A. F. W. Hamzah, I. Ismail, M. K. Yusof, S. I. M. Saany, A. Yacob, "Using Learning Analytics to Explore Responses from Student Conversations with Chatbot for Education", International Journal of Engineering Pedagogy, 2021, Vol. 11, No. 6, pp. 70-83

[30] A. O. Erkimbaev, V. Y. Zitserman, G. A. Kobzev, & A. V. Kosinov, "Standardization of Storage and Retrieval of Semi-structured Thermophysical Data in JSON-documents Associated with the Ontology", 2017.

[31] D. C. Fraga, F. Priyatna, Santana- I. S. Perez, & O. Corcho, "Virtual statistics knowledge graph generation from CSV files", *In Emerging Topics in Semantic Technologies*, 2018, pp. 235-244). IOS Press.

[32] C. Ge, Y. Li, E. Eilebrecht, B. Chandramouli, & D. Kossmann, "Speculative distributed CSV data parsing for big data analytics", *In Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 883-899.

[33] F. I. Sapundzhi, & K. Cenov, "Application of a content management system for bioinformatics websites", 2020

[34] A. V. Veremeychik, & O. A. Lapko, "Markup Languages", 2021.

[35] P. Esmailzade, "Responsive images in HTML5: A standardized solution in markup language", 2018.

[36] C. Lockard, P. Shiralkar, & X. L. Dong, "Openceres: When open information extraction meets the semi-structured web", *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Vol. 1, 2019, pp. 3047-3056.

[37] C. Lockard, X. L. Dong, A. Einolghozati, & P. Shiralkar, "Ceres: Distantly supervised relation extraction from the semi-structured web", 2018.

[38] G. Vonitsanos, A. Kanavos, P. Mylonas, & S. Sioutas, "A nosql database approach for modeling heterogeneous and semi-structured information", *In 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2018, pp. 1-8, IEEE.

[39] S. Madhusudhanan, S. Jaganathan, & J. LS, "Incremental learning for classification of unstructured data using extreme learning machines", 2018, *Algorithms*, Vol. 11, No. 10, 158.

[40] A. Malik, A. Burney, & F. Ahmed, "A Comparative Study of Unstructured Data with SQL and NO-SQL Database Management Systems". *Journal of Computer and Communications*, Vol. 8, No. 4, 2020, pp. 59-71.

[41] S. Kolhatkar, M. M. Pati, M. S. Kolhatkar, & M. M. Paranjape, "Emergence of Unstructured Data and Scope of Big Data in Indian Education", *Emergence*, Vol. 8, No. 1, 2017.

[42] D. Zappa, M. Borrelli, G. P. Clemente, & N. Savelli, "Text mining in insurance: From unstructured data to meaning", *Variance Journal*, 2021, 26122.

[43] D. Jlailaty, D. Grigori, & K. Belhajjame, "Mining business process activities from email logs", *In 2017 IEEE International Conference on Cognitive Computing (ICCC)*, 2017, pp. 112-119, IEEE.

[44] J. Dhayanithi, M. Marimuthu, G. Mohanraj, & P. Neelashkumar, "A Framework for Analysing Unstructured Data in Computing Devices", *ICTACT Journal on Soft Computing*, Vol. 12, No. 1, 2021, pp. 2464-2468.

[45] M. Lenzerini, "Data integration: A theoretical perspective", *Proceedings of the twenty-first ACM SIGMOD-SIGART symposium on Principles of database systems*, 2002, Madison, Wisconsin, USA.

[46] M. Bouzeghoub, B. F. Loscio, Z. Kedad, A. Soukane, "Heterogeneous data source integration and evolution", *DEXA*, 2002, LNCS 2453, pp: 751 – 757.

[47] P. S. Amit, & A. L. James, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", *ACM Computing Surveys*, Vol. 22, No. 3, 1990, pp: 183 – 236.

[48] L. M. Haas, D. Kossman, E. L. Wimmers, J. Yang, "Optimizing Queries Across Diverse Data Sources", *In Proc. VLDB Conference*, 1997.

[49] S. Weidman, & T. Arrison, "Improving Current Capabilities for Data Integration in Science. Steps towards Large-Scale Data Integration in the Sciences", *National Research Council of the National Academies*, 2010, pp: 1 -23.

[50] A. Y. Levy, A. Rajaraman, & J.J. Ordille,. "Querying heterogeneous information sources using source description", *Proceedings of 22th International Conference on Very Large Data Bases*, 1996, (VLBD'96).

[51] A. Y. Halevy, "Theory of answering queries using views", *SIGMOD Record*, Vol 29, No. 4, 2000, pp: 40 – 47.

[52] C. Chirathamjaree, & S. Mukviboonchai, "The Mediated Integration Architecture for Heterogeneous Data Integration", *Proceedings of IEEE TENCON'02*, 2002, pp: 7780.

[53] N. Endut, W. A. M. Fazamin, I. Ismail, M. K. Yusof, Y. A. Baker, H. Yusoff, "A Systematic Literature Review on Multi-Label Classification based on Machine Learning Algorithms", *Temp Journal*, Vol. 11, No. 2, July 2011, pp: 658 – 666.

[54] M. Macura, "Integration of Data from Heterogeneous Sources using ETL Technology", *Computer Science*, 2014, Vol. 15, No. 2, pp: 109 – 132.

[55] M. S. Jamaluddin, & N. F. Mohd Azmi, "Extraction Transformation Load (ETL) Solution for Data Integration: A Case Study of Rubber Import and Export Information", *Jurnal Teknologi*, 2015, pp: 79 – 84.

[56] G. D. Lorenzo, H. Hacid, H. Y. Paik, B. Benatallah, "Data Integrations in Mashups", *ACM SIGMOD*, Vol. 38, No. 1, 2009, pp: 59 – 66.

[57] B. Brandon, & D. G. Gregg, "Mashups: A Literature Review and Classification Framework", *Future Internet*, 2009, pp: 59 – 87.

[58] O. A. Taiwo, "Types of Machine Learning Algorithms", *New Advances in Machine Learning, Yagang Zhang (Ed.)*, ISBN: 978-953-307-034-6, InTech, University of Portsmouth United Kingdom, 2010, pp: 3 – 31.

[59] G. Kesavaraj, S. Sukumaran, "A study on classification techniques in data mining", *In Computing, Communications and Networking Technologies (ICCCNT)*, 2013; pp. 1-7.

[60] V. N. Vapnik, "The Nature of Statistical Learning Theory", *Springer*, 1995, Retrieved https://www.andrew.cmu.edu/user/kk3n/simplicity/vapnik2 000.pdf

[61] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Informatica 31*, 2007, pp. 249 – 268. Retrieved from IJS website: http://wen.ijs.si/ojs-2.4.3/index.php/informatica/article/download/148/140.

[62] I.J. Good, "Probability and the Weighing of Evidence," *Philosophy*, Vol. 26, No. 97, 1951, doi: 10.1017/S0031819100026863

[63] N.J. Nilsson, "Learning machines", *Journal of IEEE Transactions on Information Theory*, 1965, Vol. 12, No. 3, pp. 407 – 407, doi: 10.1109/TIT.1966.1053912

[64] P. Domingos, & M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss", *Machine Learning*, 1997, Vol. 29, pp. 103–130.

[65] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Informatica 31*, 2007, pp. 249 – 268.

[66] T. Hastie, R. Tibshirani, J. H. Friedman, "The elements of statistical learning, data mining, inference, and prediction," *New York: Springer Verlag*, 2001.

[67] C. Neocleous, & C. Schizas, "Artificial Neural Network Learning: A Comparative Review", *Hellenic Conference on Artificial Intelligence SETN 2002. Lecture Notes in Computer Science*, Vol. 2308, 2002, pp. 300-313, Springer, Berlin, Heidelberg, doi: 10.1007/3-540-46014-4_27.

[68] W. M. A. Fazamin, A. Noraida, Y. M. Saman, M. H. Yusof, A. Yacob, "Influence of Gamification on Students' Motivation in using E-Learning Applications Based on the Motivational Design Model", *International Journal Emerging Technolies in Learning*, Vol. 10, No. 2, 2015, pp: 30 – 34.

[69] Y. Arens, & C. Knoblock, "SIMS: Retrieving and integrating information from multiple sources", ACM SIGMOD International Conference on Management of Data (IGMOD 93), 1993, pp: 300 – 311.

[70] M. J. Carey, L. M. Haas, P. M. Schwarz, M. Arya, W. F. Cody, R. Fagin, M. Flickner, A. W. Luniewski, W. Niblack, D. Petkovic, J. Thomas, J. H. Williams, E. L. Wimmers, "Towards Heterogeneous Multimedia Information Systems: The Garlic Approach", *In 5th International Workshop on Research Issues in Data Engineering-Distributed Object Management (RIDE- DOM 1995)*, 1995, pp: 300 – 311, Taipei, Taiwan, March 6 – 7.

[71] O. M. Duschka, A. M. Keller, M. Genesereth, "Infomaster: an information integration system", *SIGMOD Rec*, Vol. 26, 1997, pp: 539 – 542

[72] P. M. D. Gray, A. Preece, N. J. Fiddian, P. R. S. Visser, "KRAFT: Knowledge fusion from distributed databases and knowledge bases", *Proceedings of the 8th International Workshop on Database and Expert Systems Applications*, 1997.

[73] E. Mena, A. Illarramendi, V. Kashyap, A. P. Sheth, "OBSERVER: An approach for query processing in global information system based on interoperation across pre-existing ontologies", *International Journal of Distributed and Parallel Databases*, Vol. 8, 2000, pp: 223 – 272

[74] K. Staken, Introduction to dbXML. XML.com, 2001.

[75] D. Suciu, "Semi structured data model", *Encyclopedia of Database Systems*, 2001.

[76] T. J. Lee, Y. Pouliot, V. Wagner, G. Gupta, D. S. Calvert, J. Tenenbaum, P. Karp, "BioWarehouse: A bioinformatics database warehouse toolkit", *BMC Bioinformatics*, Vol. 7, No. 1, 2006, pp. 170

[77] M. Xiaofeng, W. Xiaofeng, X. Min, Z. Xin, Z. Zhou, "OrientX: an integrated, schema based native XML database system", *Wuhan University J. Nat. Sci*, Vol. 11, No. 5, 2006, pp. 1192 – 1196.

[78] M. Vincini, D. Beneventano, S. Bergamaschi, "Semantic Integration of Heterogeneous Data Sources in the MOMIS Data Transformation System", *Journal of Universal Computer Science*, Vol. 19, No. 13, 2013, pp: 1986-2012

[79] D. Collarana, C. Lange, S. Auer, I. G. Gonzalez, "FuhSen: A Platform for Federated, RDF-Based Hybrid Search", *In The 16th International Conference on Web Engineering (ICWE 2016)*.

[80] M. Marjani, F. Nasaruddin, A. Ghani, S. Shamshirband, "Measuring transaction performance based on storage approaches of Native XML database", *Measurement*, 114, 2018, pp. 91 -101.

[81] F.Y. Osisanwo, J. E. T. Akinsol, O. Awodel, J. O. Hinmikaiy, A. J. Olakanmi, "Supervised Machine Learning Algorithms: Classification and Comparison", *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 48, No. 3, 2017, pp. 128 - 138.