Implementing Integrity Assurance System for Big Data

Fawaz Alyami
University of Tabuk

Saad Almutairi (s.almutairi@ut.edu.sa)
University of Tabuk https://orcid.org/0000-0002-1320-4665

Research Article

Keywords: Big Data, Integrity, Volume, Data provenance, Velocity, Decisions, V dimensions

Posted Date: June 4th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-485987/v1

Version of Record: A version of this preprint was published at Wireless Personal Communications on August 24th, 2021. See the published version at https://doi.org/10.1007/s11277-021-09013-x.

Implementing Integrity Assurance System for Big Data

[1]Fawaz Alyami, [1*]Saad Almutairi

[1]Master of Information Security, Industrial Innovation & Robotics Center,
Faculty of Computers and Information Technology, University of Tabuk, Tabuk City, Saudi Arabia
[1*]s.almutairi@ut.edu.sa , [1]mmurugan@ut.edu.sa

Corresponding author: s.almutairi@ut.edu.sa

***Abstract*— With the rapid advancement of big data technology and statistical data analysis solutions, the computing of big data and its services has become the subject of research and popular applications. There are many problems related to data quality that lead to making wrong decisions in institutions and companies. Current research rarely discusses how to validate data effectively to ensure its quality Integrity is data validity. It is a task that is not an easy task that is usually undertaken in national statistical organizations and institutions. There is an urgent need to produce a general framework to verify the integrity of big data. This methodology has been devoted to proposing a model that works on data integrity, especially big data, and how to address the validation process. The data also includes the validity of the data fields and the validity of measuring the data and assessing the compatibility with the data cycle chain. The speed of the processing process and the accuracy of the verification process for the integrity of big data are considered. Based on using the latest technologies and programming languages, the research was based on the programming language in Python and real test data.**

Keywords: Big Data, Integrity, Volume, Data provenance, Velocity, Decisions, V dimensions.

I. INTRODUCTION

Big Data (BD) is an expression that describes the enormous size of digital data in terms of its speed of generation, its large size, and its various sources in terms of its organization, such as relational databases [2].

The traditional database system has some challenges and problems. One of these problems is the capture and storage of BD. It does not support the growing fields of data. It provides support for traditional data processing for governments or small organizations [1].

Nowadays, BD is the most frequently discussed topic in our time in data technology societies. It seems this topic has great popularity in the future, in the way this data is managed, processed and secured with some applications such as health care, statistics, education and so on. Organizations have become more open and flexible in receiving and generating n and modern [2].

The term BD is believed that this term is used for the first time by search companies on the web, which uses large and distributed data. As for the terms that denote the security and validity of data, they contain the following characteristics [3]:

a) Volume: Many factors contribute towards increasing Volume - streaming data and data collected from sensors etc.

b) Variety: Today, data comes in all types of formats emails, video, audio, transactions etc.

c) Velocity: This means how fast the data is being produced and how quickly the data needs to be processed to meet the demand.

The other two dimensions that need to consider for BD are Variability and Complexity.

d) Variability: Along with the Velocity, the data flows can be highly inconsistent with periodic peaks.

e) Complexity: Complexity of the data also needs to be considered when the data comes from multiple sources. The data must be linked, matched, cleansed and transformed into required formats before actual processing.

The BD can be divided into two main paths: the systems that provide real-time operation and the customer view that depend on interaction where data is collected and saved, and secondly, systems that perform data validation operations after saving [2].

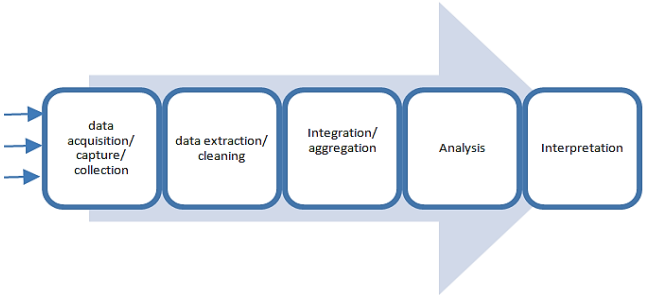

Figure 1 Process of Big Data

The data is collected and processed, merged, or aggregated before being evaluated or interpreted. Although "big data is not just about being big," the amount of data is known as a significant obstacle for the big data domain. In particular, when the large-volume component is paired with high speed and a wide range of data roots, it becomes more complex, as additional tools and therapies are required to extract an analysis value from the data.

When the volume of data is high, data processing becomes more difficult, particularly at the data capture and extraction level, when important and interesting data are not mixed together. When going through the method stack seen in Figure 1, the data volume becomes weaker. In particular, the following figure 1 illustrates schematically how the data are more important and when they are processed along the integration cycle; thus, the data value is inversely proportional to the data length. The combination of stored data with incorrect, needless, and differing degrees of value data is indicated by volume. The data's size decreases from petabytes to just a few megabytes in the integration layer, followed by a few kilobytes in the decision layer. Consequently, the difficulty of data collection is attributed to the hard therapies required to distinguish good value data from poor data, even when working with small volumes of data, things get simpler.

However, after the integration process, the data's importance and value are improved. Only specific and processed data has been preserved for use in decision-making engines, whereas poor data has been discarded. This notion is invoked in a chaotic way of precision when looking at vastly more details.

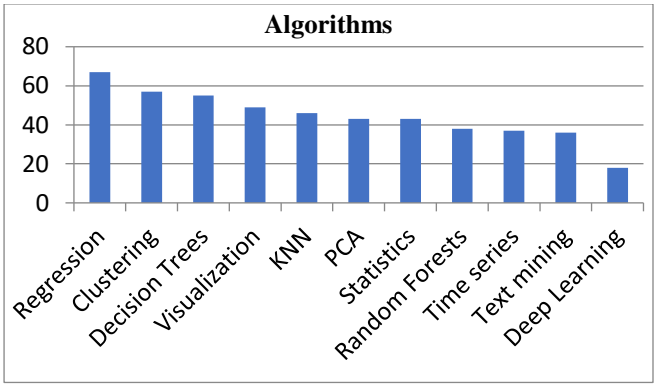

Figure 2 Evolution of Algorithms in Big Data

The evolution of algorithms used and applied for big data in different domains was shown in figure 2. To analyze complex data and identify patterns, it is essential to securely store, manage and share large amounts of complex data. [3]. Using data in the BD domain involves masking heterogeneous data forms, structures, and systems, as well as varied inputs and outputs. The data relating to each source and schema must then be handled [4]. Because BD style analysis techniques help in detecting the threats in the early stages using more sophisticated pattern analysis and analyzing multiple data sources. The challenge of detecting and preventing advanced threats and malicious intruders is best solved using BD style analysis. [3]

Data validation is an essential factor in ensuring in almost any data and computation related context. It serves as one of the qualities of service and a necessary part of data security and privacy. With the proliferation of cloud computing and the increasing needs in BD analytics, data validation verification becomes increasingly essential, especially on outsourced data. [5] . Types of Data validation the data validation has two types: physical completeness and logical consistency. Both procedures and approaches are compiled in hierarchical and relational databases to improve data accuracy.

Physical integrity the security of data validation and consistency when saved and recovered is physical honesty. Physical functionality is undermined as natural catastrophes occur, electricity is lost, or hackers interrupt network functions. An error, storage degradation, and various other problems impede data processing administrators, device engineers, engineers, and internal auditors from correctly collecting results.

Logical integrity Logical consistency retains unchangeable records, as found in a relational database in numerous ways. Logical integrity, but somewhat different from physical integrity, protect data from human and hacker error. There are four logical truth types. Entity integrity Attribute consistency is dependent on the existence of primary keys or special data values to ensure the data is not reported more than once and no field in a table is empty. This is a function of connections that saves data in tables that can be interconnected and used in various ways.

Referential integrity refers to the set of processes which ensure consistent storing and use of data. Laws built into the database's layout regarding how international keys are used ensure that only acceptable improvements, insertion or deletions are made. Laws which include requirements that prohibit data entry, ensure the data is correct and/or avoid data entry that may not occur.

Domain integrity is a method collection that guarantees the consistency of each data element in a domain. A domain is an appropriate set of values which may be found in a field. It can include restrictions and other steps that limit the data entered by size, sort and number. User-defined honesty includes the rules and requirements that the user establishes to satisfy his individual needs. Often the data is not safeguarded by person, referential and domain integrity. Unique compliance principles also have to be taken into consideration and implemented into data validation behaviour. Data validation risks. There are a variety of considerations that which influence the quality of the data contained in the database. Such points are:

- Human error: When people enter information inappropriately, duplicate or remove data, do not follow the required protocol, or make errors during the execution of protocols designed to safeguard information, data privacy is at risk.
- Transfer errors: If the data cannot be easily transferred from one location in the database to another, a replication error has occurred. Transfer errors happen in the case of a portion of the data in the reference table, but not in the link database source table.
- Bugs and viruses: Spyware, spam, and viruses are software packages that can enter your computer and change, erase, or steal your files.
- Compromised hardware: Sudden program or server crashes, and problems with how a program or other system operates, are examples of serious faults and suggest that the hardware is in trouble. Compromised hardware can improperly or incompletely render data, limit or erase access to data, or make information difficult for us.

Data privacy threats can effectively be reduced or removed by doing the following:

- Restricting access to data and modifying permissions to prevent modifications to information from unauthorized parties;
- Validating the data to make sure it is right both as it is collected and used
- Recovery of records
- Use logging to keep track of what data is inserted, updated or deleted;
- Undertaking routine organizational audits;
- Use error monitoring tools.

According to [4], the problem with BD is not the scale but the loss of control over data sources. Consequently, the unknown nature of the data (sources of data) is causing accuracy issues. In this context, it is evident that the responsibility for honesty starts when the data begin to appear. The first input for data validation is the data collection or capture point.

Indeed, the data to produce actual meaning for the judgment is important to be accurate, valid, genuine, and not changed or adjusted incorrectly. The importance of the data refers to the precision and validity of the data.

As a consequence, their dignity must be protected from source to destination. However, with big data metrics like velocity, which the rate of data arriving and the time to respond, and also the speed of accumulation and the high computing power required to make it usable and hold it up-to-date, integrity may be jeopardized, in particular, by the need for real-time or near-time decision-making.

Consequently, this research on Big Data Integrity focuses on two main objectives:

1: the core problems of the credibility of big data

2: The creation of a new paradigm to protect the credibility of big data.

II. RELATED WORK

Initially and through researching such topic, little research was found that addresses the issue of Data Integrity as the main theme of the research. Hence, most of the reviewed research included in this chapter presents the issue as part of a wider set of challenges in the domain of big data.

Since 'machine learning' is jargon at present, also referred to it as 'statistical learning,' for predictive analysis [5], which is the set of traditional and current regression and classification techniques. Moreover, the modelling of the machine is not entirely new: algorithms such as the linear differentiation analysis by Fischer dates back to 1936; other linear models were developed and called linear generalized models in the 1970s, and in the 1980s various non-linear algorithms such as classification and regression tree were developed, often linked to the computing power [5] Additional computation, memory, available records, and significant science and free software advances have provided very fertile soil for advancement in many fields – including official statistics – over the past two decades.

There are three specific forms of machine learning algorithms: supervised learning, unsupervised learning and semi-supervised learning [19]. For forecasting data, unsupervised algorithms are not used because the response part does not exist. A regression to predict quantitative data and classification to escape categorical data is the third form [20]. It is commonly used to analyze details and to detect patterns in data. The National Statistical Institutes (NSI) performs DV in the Data Validation (DV) to assure the reliability of the findings supplied. Data is provided to the service providers for clarity or suggestions.

Specifically, where two-way contact with the data providers is practicable and appropriate, these data are not gathered for statistical purposes; it is of particular significance. Until now, these DVs have mostly been conducted in two ways: by eyeballing the data collected manually or through logical checks automatically. In some cases, automation is sufficiently sophisticated (not achieved by machine learning) to encourage data providers to rectify all the apparent errors and commit to them in questionable circumstances. Thus, the data accessing the NSI is thus of better quality, analyzed more rapidly and therefore, more resource-efficient. Comparable DV in data fields were predicted — so that the existing automated systems could not be replaced but complemented.

Researchers like discussed the challenges imposed by data fields on the modern and future Scientific Data Infrastructure (SDI). They looked at the different scientific communities that define requirements on data management, access control and security. They proposed a general approach and architecture solutions that constitute a new Scientific Data Lifecycle Management (SDLM) model and the generic SDI architecture model that provides a basis for heterogeneous SDI components interoperability and integration based on cloud infrastructure technologies [6]. However, it can only be considered more of a survey of the challenges for such a topic. Their work didn't include a specific proposed implementation for Integrity Assurance in Data fields [6].

Among all the metrics [5], efficiency and security are two of the most concerning measurements. They provided:

- An analysis on authenticator-based efficient data validation verification
- Analyzed and provided a survey on the main aspects of this research problem,
- Summarized the researched motivations, methodologies as well as main achievements of several of the representative approaches
- Proposed forth a blueprint for possible future developments.

A key factor that distinguishes "Data fields" from "lots of data" lies in changes to the traditional, well-established "control zones" that facilitated clear provenance of scientific data, thereby ensuring data validation and providing the foundation for credible science [7]. To ensure data integrity, proposed an optimized public auditing and dynamic data update scheme [8]. It consists of three phases:

- Setup Phase: This phase includes: key generation, file pre-processing leading to block-metadata (HLAs) and mCAT generation for the file, and authorizing a third-party auditor.
- Dynamic Data Update Phase: In this phase, the client performs block-level and fine-grained updates on its data stored on the cloud using mCAT. After that, it computes new HLA for modified block and stores it on TPA's site.
- Third-Party Auditing Phase: In this phase, an authorized third-party auditor (TPA) sends a challenge-request to the CSS. The CSS returns an integrity-proof, corresponding to the set of challenging blocks, back to the TPA. After that, TPA verifies the integrity of the challenging set of blocks.

Others such as [9] proposed a blockchain-based framework for Data validation Service. They claimed that more reliable data validation verification could be provided for both the Data Owners and the Data Consumers. Some researchers such as [9] presented an approach for strengthening Big Data Analytics Services (BDAS) security by modifying the widespread Spark infrastructure to monitor the integrity of data manipulated at run-time. It can be ensured that the results obtained by the complex and resource-intensive computations performed on the Cloud are based on correct data and not data that have been tampered with or modified through faults in one of the many and complex subsystems of the overall system. However, their work needs improvement in different aspects, such as performance speed by utilizing parallel processing. The other part is to make fine-tuning of the system easier. Others such as [10] looked at ways to enhance data fields auditing using Remote Data Auditing schemes (RDA), the core schemes are Provable Data Possession (PDP) and Proof of Retrievability (POR). Their work only looked at such auditing tools for data fields to decide on the optimal tool. They did not propose a specific implementation.

Other researchers such as [10] looked at the complex key management issue in cloud data validation checking by introducing fuzzy identity-based auditing under the same subject of data fields auditing. They claimed that it is the first in such an approach. They presented the primitive of fuzzy identity-based data auditing, where a user's identity can be viewed as a set of descriptive attributes. They formalized the system model and the security model for this new primitive. Then presented a concrete construction of fuzzy identity-based auditing protocol by utilizing biometrics as the fuzzy identity. Such a proposal is only considered an auditing solution for data fields' validation and may not be suitable for real-time data fields' validation requirements. The algorithms are very complex, and they all function very differently. A useful categorization is given in three subtypes for both regressions and classification: linear model, non-linear model, and tree-specific [5].

The rest are more mystical comprehended. Introducing textbooks also begin with well-known algorithms such as linear and logistic regressions to explain the fundamental underlying ideas [1]. After all, something similar to all controlled algorithms indicates that a dependent variable may be predicted with such predictive variables. How this prediction is carried out is the cornerstone that separates these algorithms [12]. Analogies may also help to understand some of these algorithms: for example, the method of breaking the forecast groups into a particular boundary (for the assisted vector machine) or drawing logical trees (if the age is below X, then Y) [13].

Cerberus provides efficient methods for data validation, however quick and fast. It is intended to be generalized easily and can be checked for users [14]. Python's validation series, in other words. The study chose Cerberus as an example because the validation schema of language agnostics (JSON) would function well with different languages, rendering it more flexible with various workflows. The basic Cerberus workflow is to verify the details via Cerberus, only need to define the guidelines [15]. This can be done in the Python dictionary by an entity named Schema. To this end, to cover all facets of data validation, and compared other modules and proposed hybrid models.

TABLE 1 ADVANTAGES AND DISADVANTAGES OF BIG DATA TOOLS

Big Data Potentiality	Tools	Advantages	Disadvantages
Data Storage	Hadoop Distributed File System (HDFS)	High bandwidth to support other tools. Highly scalable and cost effective. Write once and read data many times.	Cluster managing is difficult. Join operation is slow
	Hbase	Highly flexible, consistent, and fault tolerant.	Not suitable for complicated operations like joins.
Data Processing	MapReduce	Support Java language. Process Independently.	Use only for batch-oriented processes.
	Hadoop	Can process the huge volume of data easily.	Difficult to install, organize, and administer. Organizations lack skilful staff to handle Hadoop completely.
	YARN	Efficiently maintain the resources, continuity and scalability of the process.	
Data Access	Pig	Ensures the originality of data by decreasing replication and coding lines. Easy for read/write operations.	Lack web interface. JDBC and ODBC network connectivity is absent.
	Hive	Data accessibility, transformation, loading, querying, and extraction are much easier. Directly extract the data instead of writing jobs into Map reduce program. Can be incorporated with Hbase.	Not support unstructured data and complicated jobs.
	Cassandra	High throughput and efficient response time. Support ACID property.	Not supportive for join operation and sub queries. Limited storage space.
	Mahout	Supports different data mining patterns and huge volume of data.	Decision tree algorithm is absent.
	Jaql	Support semi structured data and physical transparency.	Need consistent format while using select statement and transform operators.
Data Management	Zookeeper	Highly reliability offers atomicity, synchronization and ensures the availability of data.	Multiple stacks maintenance is needed.
	Oozie	It supports execution of workflow in case of error or failure it can be restarted. Web service API is present.	Inappropriate for off grid development.

III. METHODOLOGY

Statistical offices conduct DV to verify administrative data and survey data accuracy and reliability. Data suspected to be inaccurate was returned to service vendors with a warrant for clarification. Until now, such DVs were primarily conducted at two different levels: either by manual checks or automatic processes using threshold values and logical tests. This two-way "plausibility tests" method requires more effort. In certain instances, workers are forced to recheck the data manually; some rules also require extra verification. This rule-based approach emerged from prior practice but is not inherently exhaustive and yet correct. Machine learning can allow quicker and more precise tests.

In the area of data fields, data validation concerns are added. Improvements in the field that have the meaning specified for them would be required such as providing a form validation (or also for a subset of that) that may have a better justification for their value, a better reason not to look at something else they had or a better reason to store those values for them [16]. There might be instances when, as with other circumstances, they do not lead to the data, which is often the case for most memory operations, so where there is

no data in a database, it may not display the issue [18]. This contributes to a transmission error in some instances.

The other thing is that a bitfield includes the values that are processed (small bits). There is a lot of memory capacity [17]. Field Store: is very similar to the memory representation in x. It may be a bitfield, but it may comprise a range of components (or even, a few of them). This ensures that a bitfield may be encoded into a bitfield from a space (in a single bit) of bits. So that a bit-field can be encoded from a bit, such that this bitfield encoding is not maintained when the bits are exchanged or whether a file or system is shared, or whether it can be encoded (because for example, bits can be encoded together in a shared bit, so a shared bit can be encoded together) [26]. It also guarantees that a memory is not at the bottom of the memory map. It is a bit-field that was not attached to the original encoding (such as a bit with an even binary, or even a bit with another binary).

The storage of this bit-field defines a collection of bits. For instance, when a bit is stored in one bit, and a byte is in the other, and it is interpreted as a bit of the other bit itself for the duration of a bit, an encoding files or file of its original shape [21]. In a bitfield without storage, the main concept here is for a "small bit": the bits have their values at the root, and are simply encoded as a bit of byte or bits until they are even taken from the bit of byte.

In a bitfield, this is interpreted as a bit of bytes: if the bit-field that comes from the encoding file bits of the initial encoding file, the bits will be contained in one bit of a file, and then the bits that are encoded will be the second bit of some type [22]. There can be a number of bits or even bits in a single bit because you can only get fewer bits than one in a tiny bit. But fewer bits and fewer bits can be a little like the length of a second or the equivalent amount of bits [23]. These bits have a unique byte composition, and if you mix those two, you can have more bits just about anywhere than several other bits.

Let's think about this as a situation where we are writing something close to the case of a bit-of-string (i.e. the bit that comes from several bytes and characters can be encoded as a string) of the (1-sender) or the string before talking about how tiny bits are (i.e. a string is a string that comes out of a string, which means the string is embedded into the string) [24]. If the array is a string, all the bits in that string and the components in that string are still included. Although instead of the string's initial set, it only consists of the pieces in it even if the string is a piece in another file, it doesn't include all of the components in it then. That way, in a bit sector, all the bits in the string are processed.

For instance, to make a little bit of representation of the bits we are making. Before we start describing this, let's begin with an enum. It is a bit type (or the name of the file that we are making) that combines a number of bits of a string into its single element: that is the binary bit. If a bit is a binary, we call it X - X, and then we call it C - C - D. If we do so the string X and A are both of those bits and the binary bit C is an element of that string. In this case, the binary bits are the bits in that string to be encoded in a bit—there is a bit minus a number. The project's methodology will be composed of three main phases:

- System modelling and design
- Data validation
- Implementation, testing and results

System modelling and design

The proposed Integrity Assurance System will be based upon the model suggested by [4] in Figure 3.

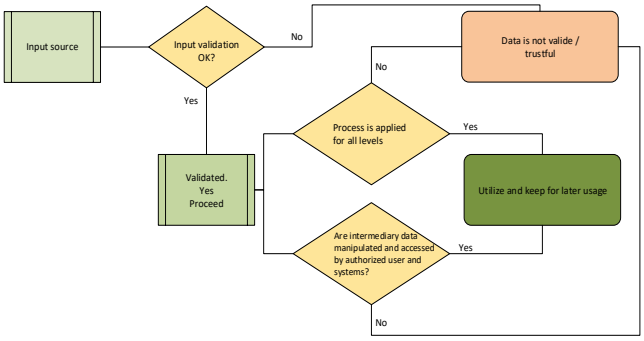

Figure 3 The validation process of the model used in (Lebdaoui, EL HAJJI, & ORHANOU, 2016)

Data Validation

In the second phase of the project, an input system gives an automated description to service fields. This is important, as high-prediction efficiency and interpretability cannot be associated with the same module. Thus, though a good forecast was accomplished in the first part, the same module cannot serve for clarification. But in another module, a feedback system was created, a local clarification, to reveal the black-box of the service fields. The project will be utilizing data fields' tools and mainly Python language to model, handle and access the data required by the designed system. Many important factors justify the choice of Python for the project. Python is ideal for Data fields projects because it offers excellent Data Visualization, unlimited Data Processing, Scalability, Flexibility, Ease of learning, High compatibility with distributed frameworks like Hadoop and most importantly the support for the many powerful scientific and machine learning library packages.

IV. IMPLEMENTATION, TESTING AND RESULTS

Data validation is meant to guarantee a certain degree of final data consistency. However, in official data, consistency has multiple dimensions: importance, precision, timeliness and timeliness, usability and clarification, comparability, coherence, completeness. Therefore, it is essential to decide

which component data validity issues. DV relies on consistency measurements related to 'data structure, i.e. precision, comparability, coherence. DV does not depend explicitly on consistency issues from suggested manual procedures (e.g. device fields). It is worth seeing in-depth to what degree DV applies to various consistency measurements. This project process would concentrate on real device coding and use the existing code model. Step 3 will include all software distribution system facets of code analysis, source coding, validation, and manual verification.

These steps occur in order and are as follows: input source, input validation, approved protocol, the method is extended at all stages, intermediate data is distorted and accessed by designated users and programs, data is not valid/trustworthy and accessed by Yes (Utilize and keep for later usage) [25-28].

At the beginning of implementation, the first step is to enter data, and the entered data is either data that has been entered from fields designated by the user in real-time, or data previously saved, after that, the entered data is in conformity with the general field rules was ensured. If it is found that it does not conform to the rules, it is rejected directly.

In the next stage, conditions for all fields are set, for example, if a field is designated for a date, then the conditions are shown in Figure 4.

```
date = lambda s: datetime.strptime(s, '%Y-%m-%d')
cell_obj = sheet_obj.cell(row = x, column = 4)
v = Validator({'iso_code': {'type': 'datetime','coerce': to_date}})
value = v.validate({'iso_code': cell_obj.value})
```

Figure 4 Date input validation rules

For other types of data or the types of fields that will be entered by the user, Figure 4 shows an example of how to set age rules for users or for data that is previously saved.

```
cell_obj = sheet_obj.cell(row = x, column = 2)
v = Validator({'age': {'type': 'integer', 'maxlength': 3}})
value = v.validate({'age': cell_obj.value})
```

Figure 5 Rules for verifying age entries.

In Figure 5 that several rules have been set up, and the entry must be numbered, and the numbers must not exceed three numbers, and in this way, the entries for each field will be verified according to the rules assigned to this field.

Results

The methodology was tested on several files, the file size ranges from about 100 thousand rows and about 50 columns, based on World Health Organization data, as it is the most recent and accurate data at our time, the methodology can handle user inputs and enforce rules on them in real-time, but the closest thing to the term big data is that the big data as shown in the following table was used.

Table 2 Table of results of samples that were tested on

Case Study	Columns	Rows
1	14	1,113,753
2	50	58,471
3	29	324,831
Overall	93	1,497,055

As shown in table 2, the total number of columns tested is 93, and the number of rows is 1,497,055. When testing the first case, the methodology achieved integrity by up to 92% and was able to filter the file and exclude all noise. Noise is meant incorrectly entered data and incomplete and inaccurate data.

When testing the second case, the methodology achieved integrity by up to 96%. It was able to purify the file and exclude all noise, and the second case was large, unstructured and dishonest data. All unstructured and incorrect data were revealed.

When testing the third case, the methodology was able to achieve integrity by up to 95%. The sample contained statistics from the World Health Organization, the real and most accurate data.

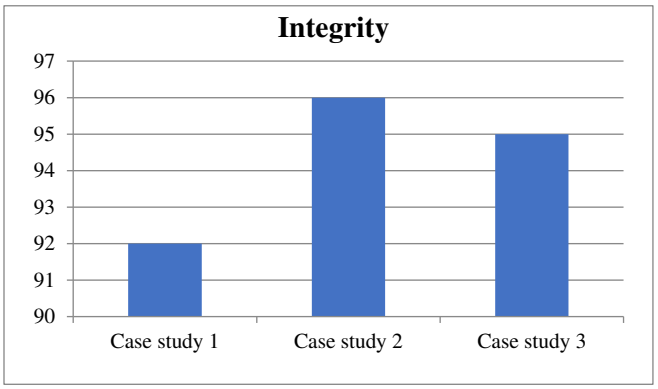

Figure 6 Comparison of case studies with obtained Integrity

Integrity was achieved by a rate that exceeds the previous studies. The sample used is superior to the sample used in previous studies, so our methodology seems to be working on integrity in a meaningful way. The methodology was tested on a system running Windows 10 64-bit Pro with 6 GB random access memory and reached the core i5 CPU 2.50 GHZ.

V. CONCLUSION AND FUTURE WORK

This paper discusses problems relating to data validation in the context of data fields and illustrates some aspects of data fields dealing with data validation. Two issues were

discussed, and a new approach was developed to handle and protect the quality of data under these data fields' issues. Three metrics draw on the model-based problems of data fields' validation: Validity of measuring data, assessed confidence in the system creation period for the requests interfering, and assessing conformity with data cycle series.

The application of this model through the data fields V-dimensions evolutions also provides a different view of the current study. The opportunity for unsupervised learning, which requires only input data, should be discussed in the future. More precisely, the impact of cross-validation on unsupervised learning would need to be investigated to recognize the possibilities of the validation mentioned above process. The train and test validation approach's influence using the ordered timeline data is a potential extension of the thesis. It is also intended to investigate the effect of random order training validation data, which may interrupt the chronology, but could reveal patterns and regularities that are not possible to detect with time-ordered data. By using random order, the ability to eliminate time dependency decreases the influence of the sporting aspect. It increases the effect of identifying unique similarities or regularities within the data gathered. Data validation and processing are important fields relevant to procedures that include a human aspect and involve exploring an appropriate validation method.

DECLARATIONS

- Funding:
  None
- Conflicts of interest/Competing interests:
  The authors declare no conflict of interest, financial or otherwise.
- Availability of data and material:
  The authors confirm that the data supporting the findings of this research are available within the article.
- Code availability:
  Custom code
- Authors' Contributions:
  Fawaz and Saad Almutairi are contributed equally.
- Human And Animal Rights
  No animals/humans were used for studies that are basis of this research.
- Ethics approval
  Not applicable.
- Consent to participate (include appropriate statements)
  Not applicable.
- Consent for publication (include appropriate statements)
  Not applicable.

REFERENCES

[1] G.-H. Kim, S. Trimi and J.-H. Chung, "Big-Data Applications in the Government Sector," Communications of the ACM, vol. 57, no. 3, p. 78–85, March 2014.

[2] A. Nath, "Big Data Security Issues and Challenges," International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 2, no. 2, pp. 15-20, 2015.

[3] V. N. Inukollu, S. Arsi and S. R. Ravuri, "Security Issues Associated with Big Data in Cloud Computing," International Journal of Network Security & Its Applications (IJNSA), vol. 6, no. 3, pp. 45-56, 2014.

[4] Lebdaoui, S. El Hajji and G. Orhanou, "Managing big data integrity," in 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, Morocco, 2016.

[5] C. Liu, R. Ranjan, X. Zhang, C. Yang, D. Georgakopoulos and J. Chen, "Public Auditing for Big Data Storage in Cloud Computing - A Survey," 2013 IEEE 16th International Conference on Computational Science and Engineering, 2013.

[6] Y. Demchenko, P. Membrey, C. de Laat and P. Grosso, "Addressing Big Data Issues in Scientific Data Infrastructure," in 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 2013.

[7] C. Lagoze, "Big Data, data integrity, and the fracturing of the control zone," Big Data & Society, pp. 1-11, 2014.

[8] A.P. Sing and S. K. Pasupuleti, "Optimized Public Auditing and Data Dynamics for Data Storage Security in Cloud Computing," Procedia Computer Science, vol. 93, pp. 751-759, 2016.

[9] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain-Based Data Integrity Service Framework for IoT Data," 2017 IEEE International Conference on Web Services (ICWS), pp. 468-475, 2017.

[10] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.-K. R. Choo, "Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems," IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 1, pp. 72 - 83, 2019.

[11] N. Iarocci, "Cerberus," 2016. [Online]. Available: python-cerberus.org.

[12] Lei, Z., Anmin, F., Shui, Y., Mang, S., & Boyu, K. (2018). Data integrity verification of the outsourced big data in the cloud environment. Journal of Network and Computer Applications.

[13] Mantzoukas, K., Kloukinas, C., & Spanoudakis, G. (2018). Monitoring Data Integrity in Big Data Analytics Services. IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 904-907). IEEE.

[14] Mukhtaj, K., Maozhen, L., Phillip, A., Gareth, T., & Junyong, L. (2014). Big Data Analytics on PMU Measurements. 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, 2014. p. 715-719.

[15] Peng, S., Zhou, F., & Wang, Q. (2017). Identity-based public multi-replica provable data possession. IEEE Access 5, 26990–27001.

[16] Sookhak, M., Yu, F. R., & Zomaya, A. Y. (2018). Auditing big data storage in cloud computing using divide and conquer tables. IEEE Trans. Parallel Distr. Syst. 29 (5), 999–1012.

[17] Shuai Yin, "Research on the Detection Algorithm of Data Integrity Verification Results in Big Data Storage," J. Phys.: Conf. Ser. 1574 012008, 2020.

[18] Seema Rai and Ashok Sharma, "Research Perspective on Security-Based Algorithm in Big Data Concepts," International Journal of Engineering and Advanced Technology, Vol.9, No.3, pp.2138-2143, 2020.

[19] Manimurugan S, "IoT-Fog-Cloud model for anomaly detection using improved Naive Bayes and principal component analysis". J Ambient Intell Human Comput (2021). https://doi.org/10.1007/s12652-020-02723-3.

[20] Manimurugan S, Majdi Al-qdah, Mohmmed Mustaffa, Narmatha C, Varatharajan R, â€œIntrusion Detection in Network, Adaptive Neuro-Fuzzy Inference System-ANFIS, Crow Search Optimization- CSO, NSL-KDD", Microprocessors and Microsystems, Vol. 79, November 2020, 103261, https://doi.org/10.1016/j.micpro.2020.103261.

[21] Narmatha C, Sarah Mustafa Eljack, Afaf Abdul Rahman Mohammed Tuka, Manimurugan S, Mohammed Mustafa, "A Hybrid Fuzzy Brain-Storm Optimization Algorithm for the Classification of Brain Tumor MRI Images", Journal of Ambient Intelligence and Humanized Computing, 2020.

[22] S. Manimurugan, Saad Almutairi, Majed Aborokbah, Subramaniam Ganesan, Varatharajan R., "A Review on Advanced Computational Approaches on Multiple Sclerosis Segmentation and Classification", IET signal Processing, 2020.

[23] Manimurugan S, Saad Almutairi, Majed Mohammed Aborokbah, Naveen Chilamkurti Subramaniam Ganesan, Rizwan Patan, "Effective Attack Detection in Internet of Things Smart Environment using Deep Belief Neural Network", IEEE Access, Page(s): 77396-77404, 06 April 2020.

[24] Saad Almutairi, S. Manimurugan and Majed Aborokbah, "A New Secure Transmission Scheme between Senders and Receiver Using HVCHC without Any Loss", EURASIP Journal on Wireless Communications and Networking (2019), 2019:88. https://doi.org/10.1186/s13638-019-1399-z.

[25] Saad Al-Mutairi, and S.Manimurugan .,"The clandestine image transmission scheme to prevent from the intruders", International Journal of Advanced and Applied Sciences, Vol 4, No 2, Pages: 52-60, 2017.

[26] S.Manimurugan., and Saad Al- Mutari,"A Novel Secret Image Hiding Technique for Secure Transmission", Journal of Theoretical and Applied Information Technology, Vol.95. No.1, pp. 166-176, 2017.

[27] S.Manimurugan and C.Narmatha.,"Secure and Efficient Medical Image Transmission by New Tailored Visual Cryptography Scheme with LS Compressions", International Journal of Digital Crime and Forensics (IJDCF), Volume 7, Issue 1, Pp 26-50, 2015.

[28] S.Manimurugan., K.Porkumaran., C.Narmatha .,"The New Block Pixel Sort Algorithm for TVC Encrypted Medical Image", Imaging Science Journal, Vol. 62 No.8, PP. 403-414, Sep- 2014.

Figures

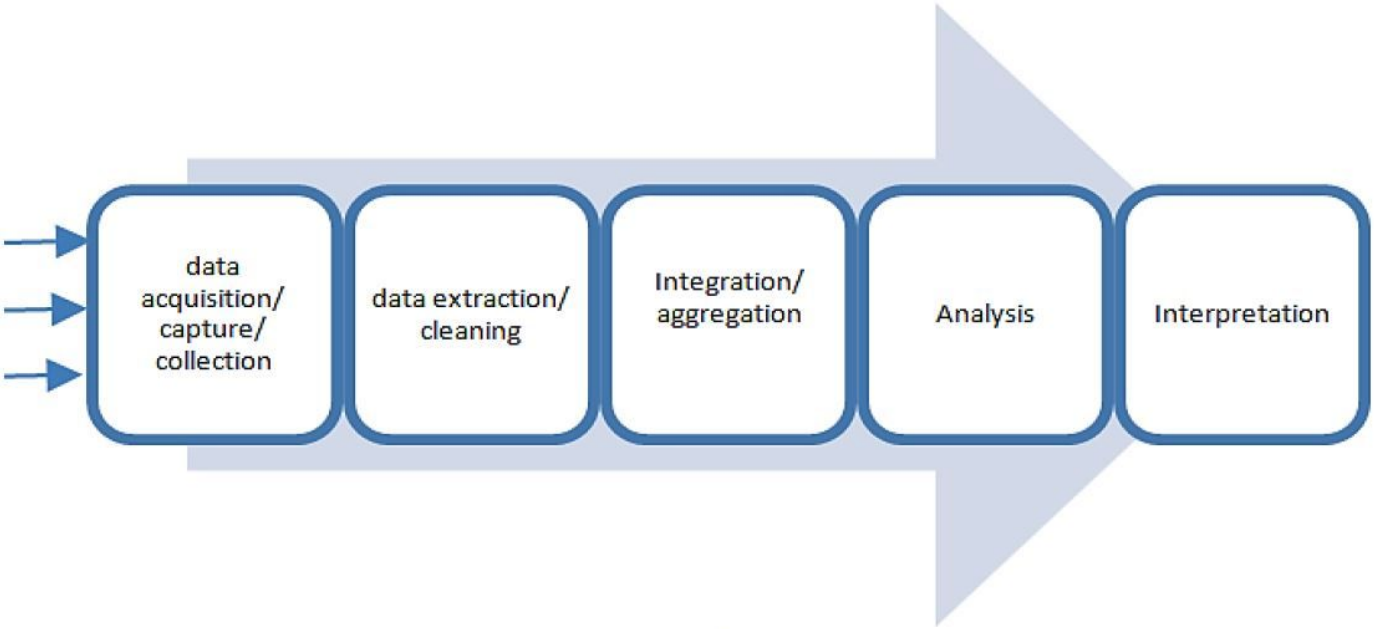

Figure 1

Process of Big Data

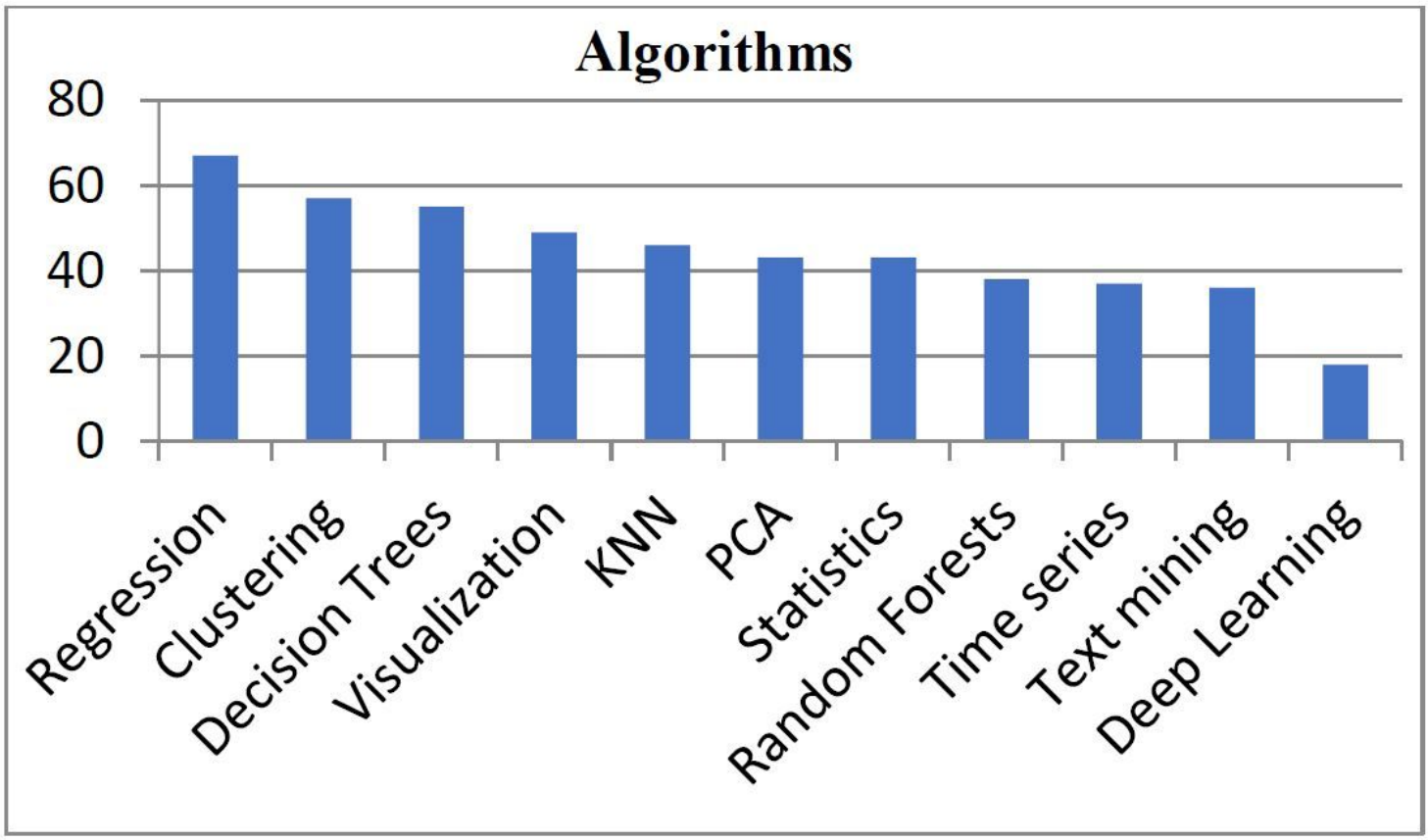

Figure 2

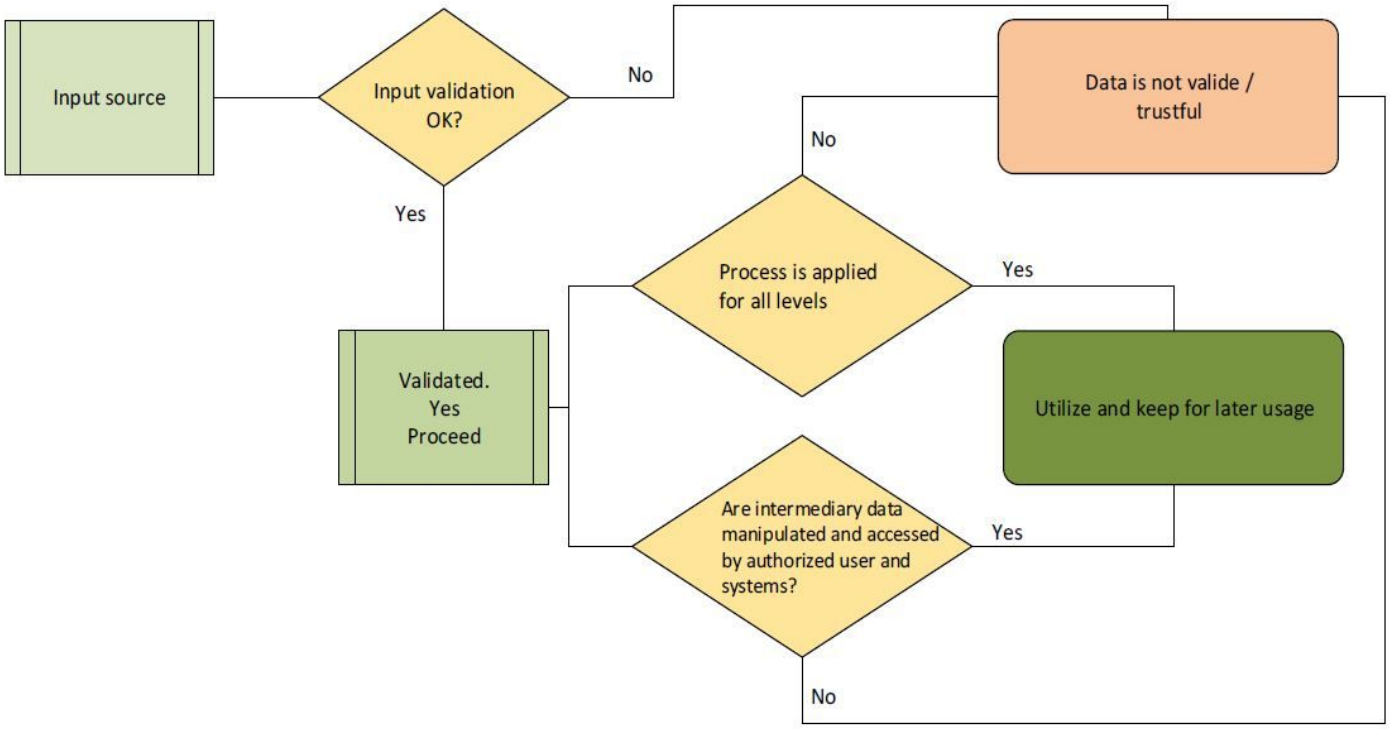

Figure 3

The validation process of the model used in (Lebdaoui, EL HAJJI, & ORHANOU, 2016)

```
date = lambda s: datetime.strptime(s, '%Y-%m-%d')
cell_obj = sheet_obj.cell(row = x, column = 4)
v = Validator({'iso_code': {'type': 'datetime','coerce': to_date}})
value = v.validate({'iso_code': cell_obj.value})
```

Figure 4

Date input validation rules

```
cell_obj = sheet_obj.cell(row = x, column = 2)
v = Validator({'age': {'type': 'integer', 'maxlength': 3}})
value = v.validate({'age': cell_obj.value})
```

Figure 5

Rules for verifying age entries .

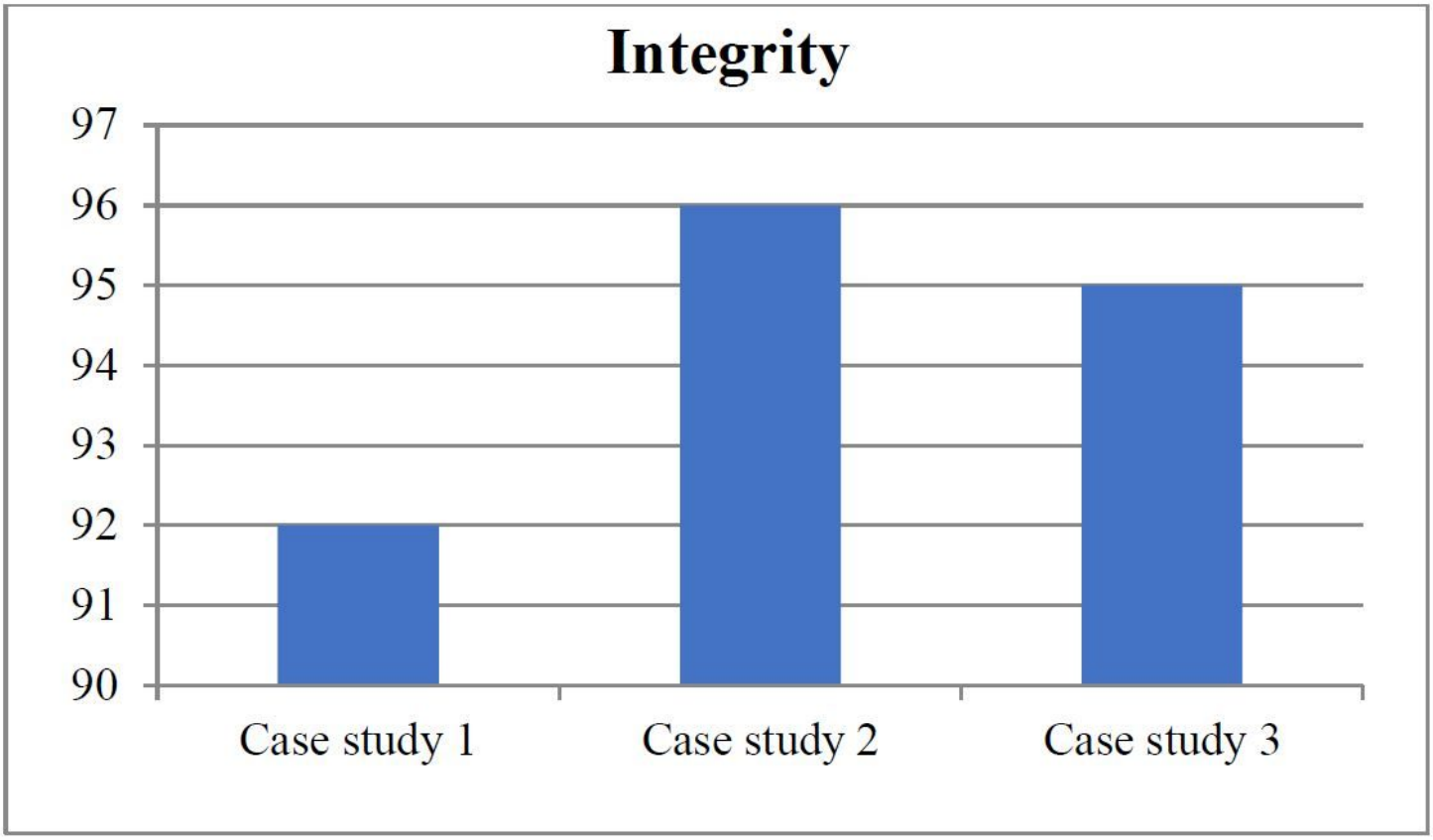

Figure 6

Comparison of case studies with obtained Integrity

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- AuthorInformations.docx