Original Research Paper

Fals-Ism: A Graph Isomorphism Framework for Multi-Level Detection of Falsified PDF Documents

[1]**Josue Nguinabe,** [1,3]**Franklin Tchakounte,** [2]**Patient Murhula Buhendwa and** [3]**Marcellin Atemkeng**

[1]*Department of Computer Science and Mathematics, Faculty of Science, University of Ngaoundere, Cameroon*
[2]*African Institute of Mathematical Science (AIMS), Limbe, Cameroon*
[3]*Department of Mathematics, Rhodes University, Makhanda 6139, South Africa*

Article history
Received: 08-01-2023
Revised: 22-02-2023
Accepted: 25-03-2023

Corresponding Author:
Marcellin Atemkeng
Department of Mathematics,
Rhodes University, Makhanda
6139, South Africa
Email: m.atemkeng@gmail.com

Abstract: Fake Portable Document Format (PDF) documents are disseminated in an incredible rhythm across social media. Negative incidences are obvious but effective solutions identifying falsified items in the PDF are still in need. Unlike determining malicious scripts inserted into the file, this research aims at identifying falsified objects from different layers of the document. Specifically, we introduce Fals-Ism, a novel approach to detect falsified PDF documents based on graph isomorphism. Each document is transformed and characterized by metadata, structure, and content required to build the corresponding graph such that any alteration is reflected on the complete graph. The graph is input to the isomorphism search algorithm namely; VF2 to verify if there is a similarity-based isomorphism. Experiments are conducted on (36) PDF documents considering metadata, structure, and content modifications. The results show that Fals-Ism (i) Is efficient to detect forgery at metadata level, structure, and content; (ii) Is robust and resistant to forgery attacks such as insertion, deletion, and modification of information; (iii) Does not require certain information about the PDF documents beforehand to perform the detection. Fals-Ism can detect different types of falsifications in PDF (version 1.7 or higher) with an accuracy of 90%. A comparison with similar work confirms that Fals-Ism could be a complementary tool for fake news detection.

Keywords: Detection, Falsification, PDF, Graph, Isomorphism, Social Media

Introduction

Portable Document Format (PDF) has become the most popular and widely used format for describing digital documents worldwide (Bradley, 2011). In recent years, social networks have allowed malicious people to distribute falsified documents, making fraud on social networks an economic issue. There are a lot of investments to fight against that. People are looking to equip themselves with powerful fraud detection tools, regardless of their field of activity (Shu *et al*., 2017; Yang *et al*., 2019; Elhadad *et al*., 2019; Tchakounté *et al*.,´ 2020a; Kaliyar *et al*., 2021). According to research firm PCW[1], 47% of companies surveyed globally said they were victims of social media fraud every 24 months in 2020 (Rivera *et al*., 2020), compared to 46% in 2022 (PWC, 2022). These statistics reveal that despite sophisticated fraud detection tools, the concern remains.

Today, physical information is digitized to facilitate its manipulation with technologies. Organizations need to archive information for easy storage and for future processing. For this, digital transformation is ensured by digitization in PDF documents, allowing distribution via means of communication. The critical issue of these processes is to guarantee the integrity of the document; digital files are easier to falsify than paper documents. This aspect should be carefully considered when designing and choosing digital file formats (Bradley, 2011). Digital files such as PDFs are one of the most secure file formats when it comes to falsification (Bradley, 2011; Laptev *et al*., 2017). This is why this format is widely used to store information.

[1] Price Water house Coopers (PCW)

Current approaches to PDF integrity primarily look for malware infiltration and document tampering. Regarding malware infiltration, the authors rely on the machine and deep learning techniques to profile normal and abnormal PDFs using images and other features (Gebhardt *et al*., 2013; Beaugnon and Husson, 2017; Khan *et al*., 2018; Ayinala and Grandhi, 2021). Regarding document forgery, Khanna *et al*. (2008); Elkasrawi and Shafait (2014); Abed (2015) attempt to identify the scanned and the printed version of a PDF document based on the use of texture analysis. With the use of cryptography, the counterfeit is identified by mechanisms that avoid escaping signatures linked to the document (Perry *et al*., 2000; Picard *et al*., 2004; Cheddad *et al*., 2008; Schouten and Jacobs, 2009; Ibrahim *et al*., 2010; Yang, 2014; Tchakounté *et al*., 2020b; Tchakounte *et al*., 2021). Research attempts have been proposed for the recognition of falsification with artificial intelligence based on imaged features extraction (Van Beusekom *et al*., 2013; Patgar and Vasudev, 2013; Bertrand *et al*., 2013; 2015; Patgar *et al*., 2014; Abramova, 2016; Laptev *et al*., 2017). While all of these works have potential, there are still shortcomings to overcome. Detecting inserted malware requires in depth dynamic analysis while manipulating the document to study the different flows and activities performed. With the heterogeneous aspects of the software and the PDF document, this is hardly feasible. But the simple variation of the content of the document modifies the signatures and the profiles of the document. The document will therefore be wrongly classified as false.

In this study, with the assumption to have the original PDF document, we propose a similarity matching approach to recognize falsified PDFs. The proposed method, namely Fals-Ism, relies on a robust theoretical and well-proven tool to solve similarity assessment problems; graph isomorphism. A document is profiled in three levels e.g., metadata, structure, and contents. Based on these profiles, we convert the target document into a graph that we compare against the original document graph, similarly profiled. With graph isomorphism matching principles, we extract exactly where the alterations are applied in the falsified PDF document. The key contributions of this study are:

- We propose a technique to transform a PDF into a multi-level graph that conserved metadata, structure, and content
- We introduce Fals-Ism, a new efficient similarity approach based on graph isomorphism, which detects PDF falsification at three levels: Metadata, structure, and content
- We conduct experiments on 36 PDF documents with multi-level variation, taking into account variations such as inserting, modification, deletion, and varying document pages. Results demonstrate that Fals-Ism is able to detect PDF (version 1.7 or above) falsification with an accuracy of 90%, which improves similar work in the literature

The manipulation of PDF documents requires a deep understanding of their anatomy. The sections give a basic idea of how to construct, structure, and secure PDF files. All the information is based on the PDF standard, ISO/IEC 32000-1:2008.

PDF Anatomy

A PDF supports eight basic data types. Each type corresponds to a specific set of values described as follows. Boolean type is represented by the keyword true or false. Number type refers to integer and real. Strings type can be characters between brackets “()” or hexadecimal data between quotation marks “<<>>”. Type names are sequences of characters with the null character as an exception. The special character “/" named slash is used to enter the name type. A type of “array" that can hold multiple object types including names, strings, and arrays. Type “dictionaries” which is similar to a dictionary containing the description followed by a word. The type description can contain objects or another dictionary. The “stream” type such as strings in programming but can have an unlimited length. Streams are a special type for holding big data that a simple “string” cannot hold. Finally, the null object is an empty object represented by the symbol null.

The structure of PDF files determines (i) How objects are stored, (ii) How they are accessed, and (iii) How they are updated in a PDF document. This structure is independent of the semantics of the objects. All PDF files have a common structure which is subdivided into 4 parts: The header, body, cross-ref table, and trailer. The header is the file header and is the first line of the source code of a single line PDF. This part contains five characters, we have “% PDF-” associated with the version number. The part of the body that contains the document content. The body represents the actual content of the data that makes up PDF documents. The cross-ref table is the most important part of the document structure. The trailer known as the final part of the file, is used to find the cross-ref table and several useful objects in the file. The table presents the different keys, their values, and types that can be encountered as end entries.

The structure of a PDF document specifies how the basic object types are used to represent its components (pages, fonts, annotations, etc.,). A PDF file’s document structure consists of a number of objects arranged in the body in a hierarchical fashion. They are arranged in a page tree according to the document catalog’s specifications, where they are divided into page objects. This catalog includes references to other objects that detail the document’s content and instructions on how that content should be displayed. It relates to a number of things, but the page tree a structure that arranges and makes all page objects accessible is the one that matters the most to us. For an intensive discussion on PDF documents, we refer the reader to Adobe Systems 2008.

Graph Isomorphism

A graph G is a pair $G = (S(G), A(G))$, where the elements of $S(G)$ are the vertices or nodes and $A(G)$ is the set of edges with $A(G) \subseteq S(G) \times S(G)$. $G_1 = (S_1(G), A_1(G))$ is a subgraph of G if $S_1(G) \subseteq S(G)$ and $A_1(G) \subseteq A(G)$.

An application f: $S(G) \longrightarrow S(H)$ is a graph morphism if the image of any edge of a graph G is an edge of a graph H. Mathematically, this is given as; if $\forall(u,v) \in A(G)$ and $(f(u), f(v)) \in A(H)$ then G and H are homomorphic if there exists a morphism between them. The application f realized an isomorphism when G and H are homomorphic and each of them is bijective, i.e., there exists a univocal relation f: $S(G) \rightarrow S(H)$ such that:

$$\left(u,v\right) \in A(G) \Leftrightarrow \left(f(u), f(v)\right) \in A(H) \tag{1}$$

which means $G \cong H$.

Materials and Methods

The proposed approach is divided into four modules. The first module is the features extraction in a PDF document, the second module consists in transforming the features into graphs; the third module search for the isomorphism between the graphs, and the last module is where the decision is made.

Mathematical Formulation

We are interested in graph isomorphism to detect a forgery in a PDF file. We consider the texts in the PDF document as a graph whose nodes are words and edges are the semantic relationship in the document. Since a PDF document can be decomposed into a graph, an isomorphism between two PDF documents confirms that both PDF documents are equal. Let G be a graph representing a PDF file, G can be decomposed into a set of n subgraphs G_1, $G_2,\cdots, G_n$ such that there exists a strict structural relationship $\Re$ between the subgraphs i.e., if G is a PDF file then:

$$\exists G_1, G_2, ..., G_n \ such \ \ that \, \Re\left(G_1, G_2, ... G_n\right) \tag{2}$$

where all the G_i are subgraphs of G and the notation $\Re(G_1, G_2,\ldots, G_n)$ means there are strict structural relationships between two successive subgraphs G_i and G_{i+1}. Two PDF files are identical if and only if there exists an isomorphism between their subgraphs. This is written mathematically as follows. Let G and H be two PDF files with subgraphs $G_1, G_2,\ldots,G_n$ and $H_1, H_2,\ldots,H_m$ respectively; G and H are identical if $n = m$ and for all $i = 1, 2,\ldots, n$ there exists an application f: $S(G_i) \rightarrow S(H_i)$ such that:

$$\left(u,v\right) \in A(G_i) \Leftrightarrow \left(f(u), f(v)\right) \in A(H_i) \tag{3}$$

The problem that rises is how to find f. There are two types of algorithms for finding isomorphisms between graphs. The first one is exact matching which includes algorithms that look for a perfect match between two graphs before considering them as isomorphic. The second one is inexact matching or fault tolerance matching including algorithms qualified as inexact because they relax constraints allowing some errors and noise when searching for an isomorphism between two graphs. To allocate a redundancy of edges and vertices, they demand that each vertex of the first graph be able to map distinct vertices in the second graph regardless of the edge orientation between the vertices (Wang *et al.*, 2018).

The algorithm for finding an isomorphism between two PDF files used in this study is the VF2 (Cordella *et al.*, 2001). In spite of the size and type of the graph that needs to be matched, the exact matching method VF2 has consistently been able to solve the isomorphism problem (Cordella *et al.*, 2004). The VF2 algorithm was proposed by Cordella *et al.* (2001) for large graph adaptation. It is a heuristic method with features inherited from the VF algorithm (Cordella *et al.*, 1998; 1999), which reduces VF memory space from $O(n^2)$ to $O(n)$ where n is the number of vertices of the graph. The VF2 checks an isomorphism as follows: For two graphs G and H, a state S_0 to initialize all data structures is set. The matching method receives an instance of the algorithm, which checks to see if two pairs of vertices originating from G and H match up. If the verification returns true then the candidate pairs are added to the set of pairs corresponding to both G and H and the procedure is repeated until all candidate pairs are tested. If a match is found then a return is made to the last candidate pair found otherwise no match is returned.

Algorithms and Flowchart of the Proposed Method

We propose an approach that includes three algorithms. Algorithm 1 is the feature extraction level which is responsible for extracting the features in a PDF document. The input of this module is a PDF document and the output is the metadata, structure, and content of the PDF. To extract different features that make up a PDF file, the pdf reader library[2] is used. This library allows access to all the features in a PDF file. The outputs of Algorithm 1 are inputs of Algorithm 2 which is responsible to transform the inputs into graphs. Algorithm 3 is the module where the isomorphism is checked. The inputs of these modules are several graphs and the output is Boolean indicating if the input graphs realize an isomorphism or not.

[2] Application Programming Interface (API) that allows extraction of text, images, and other data from PDF documents (simple or protected

Algorithm 1: Feature extraction

```
Input: PDF 1, PDF 2; /* two any PDF files*/
Output: Metadata, Structure, and Content features;
procedure FEATURE EXTRACTION (PDF files):
    Load and Open PDFs in source mode; /*Loading of files*/
    Locate the root; /*To find the catalog of the file*/
    Extract Metadata, Structure, and Content; /*Final extraction*/
end procedure
```

Algorithm 2: Transform to graphs

```
Input: Metadata, Structure, and Content Features;
Output: Metadata, Structure, and Content graphs;
Procedure Graph Transformation (Metadata, Structure, Content):
    Define graph G and its edges; /*Initialisation */
    Link the root to the Page tree; /*graph building*/
    Link page tree to other objects in the hierarchy;
    return graph of metadata, structure, and content;
end procedure
```

Algorithm 3: Isomorphism check

```
Input: Metadata, Structure, and Content graphs;
Output: PDF is Falsified or PDF is not Falsified;
Procedure Isomorphism Checking and Decision Making ($G_1$, $G_2$):

    if is isomorphic ($G_{1M}$, $G_{2M}$) == False then
        the metadata of this PDF file has been Falsified.
    Else
        if is isomorphic ($G_{1S}$, $G_{2S}$) == False then
            the structure of this PDF file has been Falsified.
        Else
            if is isomorphic ($G_{1C}$, $G_{2C}$) == False then
                the content of this PDF file has been Falsified.
            Else Return PDF is not falsified.
            end if
        end if
    end if
end procedure
```

Experimental Validation

This section evaluates Fals-Ism with several tests. Ten experiments are carried out using different sample datasets. The tests are carried out following three types of falsification attacks such as (i) Insertion of information, which consists of editing the content of the PDF document, (ii) Alteration or deletion of some information that is part of the content of the PDF document and (iii) Modification of information, which consist to change an element in the document.

Datasets

We simulate 36 samples of PDF documents and split them into 10 sample datasets. Because in real life, forgery can occur anywhere in a PDF document, to cover many of these aspects, we have chosen to take several variants of PDF documents as follows:

1. Twenty of the PDF documents contain only raw texts
2. One of the PDF documents contains texts and images
3. One of the PDF documents is generated by scanning the original physical PDF document as an image without applying Optical Character Recognition (OCR)
4. One of the PDF documents is generated by scanning the original physical PDF document as an image with Optical Character Recognition (OCR)
5. One of the PDF documents is generated by scanning a physical document as an image with Optical Character Recognition (OCR) and still contains images
6. One of the documents contains texts and tables
7. One of the PDF documents contains texts, mathematical equations, and symbols
8. One of the PDF documents contains a signature
9. Seven of the PDF documents are from PDF versions 1.0-1.6
10. Two of the PDF documents are of versions greater than or equal to 1.7

Experiments

On each sample dataset, tests are performed on three levels (validation stages) before a decision is made on the authenticity of the document. The first validation step deals with the metadata, the second with the structure, and the third validation step with the content. At each of the above steps, different types of falsification such as Insertion, deletion, and modification of information are introduced on each PDF document and submitted to Fals-Ism for detection.

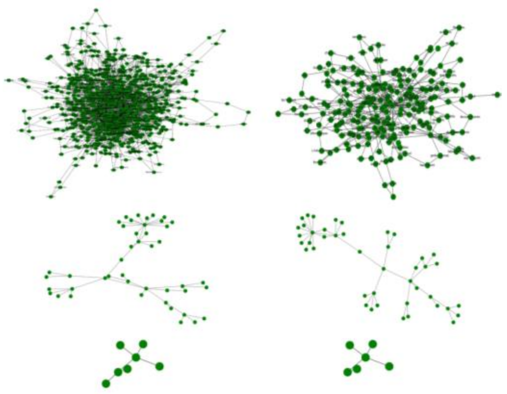

Fig. 1: Graph of content (top-left) and its falsified version (top-right); Graph of structure (middle-left) and its falsified version (middle-right); Graph of metadata (bottom-left) and its falsified version (bottom-right)

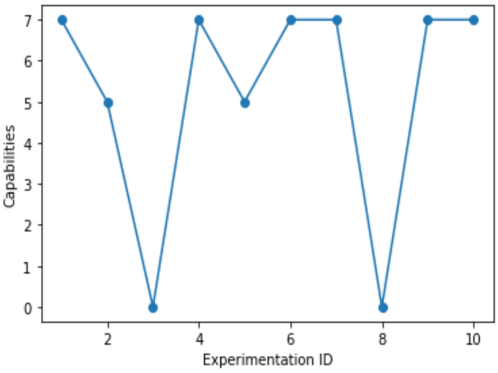

Fig. 2: Detection curve at the content level

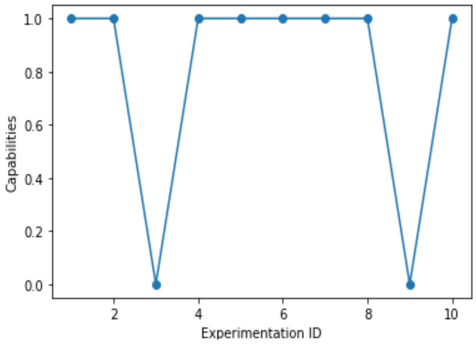

Fig. 3: Detection curve at the structure level Table 2. Results obtained on the whole samples

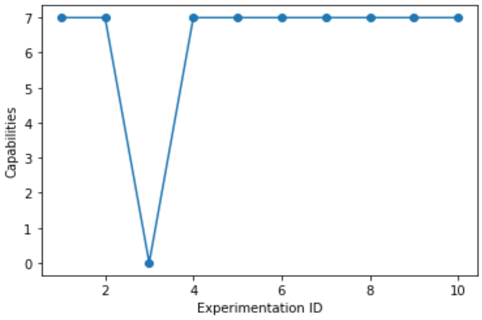

Fig. 4: Detection curve at the metadata level

Figure 1 shows the graph of content (top-left) and its falsified version (top-right); the graph of structure (middle-left) and its falsified version (middle-right); the graph of metadata (bottom-left) and its falsified version (bottom-right) of a single PDF document. The figure shows that the three graphs (content, structure, and metadata) are different from their falsified version. This situation indicates that there exists no isomorphism between them.

Figure 2 shows the detection pattern made by Fals-Ism on the content of all 10 samples of the dataset. The x-axis represents the 10 sample datasets used to test Fals-Ism and the y-axis is the system/Fals-Ism capability. The capability is expressed in three bits meaning that there are 23 possibilities in Table 1. For example, the value 7 refers to 111 in binary. This means that the system was able to detect the falsification by insertion, modification, and deletion of information on the content of the PDF document that makes up this experiment. The value 5 refers to 101 in binary. It means that the system was able to detect insertion and deletion but not modification of information. Results in Fig. 2 show that experiments 1, 4, 6, 7, 9, and 10 are falsified because all bits are 111 which implies falsification by insertion, deletion, or modification. The curve decreases slightly from experiment 2 and drops completely to zero in experiment 3. The same is observed in experiments 5 and 8.

Contrary to the type of falsification in the metadata or the content of a PDF document, the only type of falsification of the structure of a document is the falsification by modification (for example one can modify the font of the characters in a PDF document) but with no deletion or insertion, reason why we observe only one bit on the curve corresponding to the falsification by modification of the PDF document structure in each experiment, where 1 means that the detection was a success and 0 otherwise.

In experiments 1, 2, 4, 5, 6, 7, 8, and 10 in Fig. 3 the curve is constant everywhere except in experiments 3 and 9. This means that the system was able to detect all the modifications made to all the structures of the PDF documents. However, in experiments 3 and 9, the bits are 0, which means that the system was unable to detect the falsification carried out on the structure of these PDF documents.

With regard to metadata, the three types of falsification, namely, falsification by insertion, deletion, and modification are carried out on the datasets in the same way as on the content of the documents. Figure 4 represents the level of detection made by Fals-Ism on the metadata.

Table 1: Capabilities interpretations

Capabilities			
Insertion	Deletion	Modification	Interpretations
0	0	0	The system is not able to detect insertion-deletion and modification
1	0	0	The system can detect insertion but is not able to detect deletion and modification
0	1	0	The system is not able to detect insertion and modification but can detect deletion
0	0	1	The system is not able to detect insertion and deletion but can detect modification
1	1	0	The system can detect insertion and deletion but cannot detect modification
1	0	1	The system can detect insertion and modification but cannot detect deletion
0	1	1	The system can detect deletion and modification but cannot detect insertion
1	1	1	The system is able to detect insertion-deletion and modification

Table 2: Result obtained on the whole samples

Experiments	Step 1	Step 2	Step 3
1	The system has detected deletion, modification and insertion of the text in content of the document	The modification made on the structure of this PDF document	The system has detected the insertion, modification and deletion of information made the metadata of this PDF document
2	The system has detected the deletion and insertion of an image in the content of the document but cannot detect when modification is made to the structure of the image or if the image even has been replaced by another	The system has detected the modification made the structure of this PDF document	The system has detected the insertion, modification and deletion of information made on the metadata of this PDF document
3	The system was not able to detect falsifications that were made on the content of this PDF document	The system was not able to detect falsifications that were made on the structure of this PDF document	The system was not able to detect falsifications that was made on the metadata of this PDF document
4	The system detected the insertion, deletion or modification made on the content of the PDF document	The system has detected the modification made on the structure of this PDF document	The system detected the insertion, modification and deletion of information made on the metadata of this PDF document
5	The system has detected the deletion or insertion of an image in the content of the document but cannot detect when modification is made to the structure of the image or if the image in question has been replaced by another	The system has detected the modification made on the structure of this PDF document	The system has detected the insertion, modification and deletion of information made on the metadata of this PDF document
6	The system detected the insertion, deletion and modification of a table in a PDF document	The system detected modification made on the structure of this PDF document	The system detected the insertion, deletion and modification at the metadata level of this PDF document
7	The system detected the insertion, deletion or modification made on equations and symbols in the PDF document	The system detected the modification made on the structure in this PDF document	The system detected the insertion, deletion or modification made on the metadata in this PDF document
8	The system did not detect the insertion, deletion and modification of the signature on this PDF document	The system has detected a modification made on the structure of this PDF document	The system detected an insertion, deletion and modification of information made on the metadata of this PDF document
9	The system detected the insertion, deletion and modification of information the content level of this PDF document	The system did not detect the modifications made on the structure of PDF with version less than 1.7	The system detected the insertion, deletion and modification made on the metadata of this PDF document
10	The system detected the falsifications of types insertion, modification, and deletion made at the level of the contents of these PDF documents and the falsifications of types modifications made to their structure	The system detected falsifications of types insertion, modification and deletion at the level of the metadata of the PDF documents	The system detected falsifications of types insertion, modification, and deletion at the level of metadata of the PDF documents

In experiments 1, 2, 4, 5, 6, 7, 8, 9, and 10 Fig. 4, all the points correspond to 7 i.e., 111 bits in binary. This means that the system was able to detect all insertions, deletions, and modifications of information in the metadata of these PDF documents except at the level of experiment 3 where the system was totally unable to detect any type of falsification performed on this PDF document.

Table 2 shows the test results for each of the three steps and for all experiments, we carried out with the 10 sample datasets. The first column corresponds to the sample dataset. The three other columns refer to the level of experiments: Content, structure, and metadata. At each level, we apply the three types of falsification: insertion, deletion, and modification.

Results and Discussions

Figures 1-4 depict that Fals-Ism has the following capabilities:

- Fals-Ism is an effective tool for detecting PDF documents consisting only of raw text in terms of detection at the content level. The system is able to detect even if two characters are exchanged in the document
- The system is able to detect any manipulation performed by inserting, deleting, or modifying information
- The results obtained show that the proposed method is effective in terms of detecting falsification at the metadata level with a detection rate of 90%

Fals-Ism has some weaknesses as observed:

- We observe that the performance could decrease with the number of pages. Experiments with more complex and very large PDF files are required
- Detection at the document structure level remains uncertain except for PDFs with versions greater than or equal to 1.7. Other versions should be deeply investigated
- The system is completely unable to perform forgery detection against PDF documents scanned as an image or generated directly as an image
- The system is completely unable to detect a forgery made by, for example, changing a signature in a document

Comparison with Similar Work

A Decentralized Document Management System (DDMS) was suggested by Han *et al*. (2021) to increase the security of digital documents in order to secure them. For greater document security, DDMS distributes access rights to a number of users by symmetrically encrypting the document with a key and dividing the key using Shamir's secret sharing. Each split key is managed using blockchain and when the document is retrieved using a developed smart contract, the whole symmetric keys are rebuilt. The proposed DDMS can provide stronger security with a fair performance overhead. Smart contracts and blockchain technology are used by Serranito *et al*. (2020) to build a decentralized verification solution for university diplomas and other higher education qualifications. An actual blockchain is used to test a prototype of the implementation and the challenges that were faced are identified and assessed with an emphasis on how well the decentralization mechanism worked. According to the authors, the technology enables higher education institutions to store certificates they issue in the blockchain where hiring companies may verify their integrity and legitimacy. Laptev *et al*. (2017) suggested attacking the PDF file's source code to make the process of identifying modifications in PDF files easier. Results show that the method is efficient for analyzing PDF files to determine their integrity.

GraDID is proposed by Jung *et al*. (2022) to determine if a document substitutes another one. The authors studied the consistency of the body context of a document formalized as a graph of nodes where the node is taken as the whole text. Unlike our approach, this way of doing is coarse grained and not precise. Moreover, this study is limited when the structure and metadata are altered.

Kada *et al*. (2022) proposed a way to identify fake identity documents based on the reconstruction of holograms. Unlike ours, this proposal considers general holograms and therefore lacks precision.

Patil *et al*. (2022) relied on sequencing the order of pixels to discover irregularities in handwritten documents. This type of document is not within the scope of this research.

Methods based on blockchain technology offer a very good level of security, but the implementation of this technology is very complex and requires a lot of resources in terms of machine power, storage space, etc. All these parameters imply a huge financial cost, which is why in Ali and Bhaya (2021) the implementation of this solution has not yet been deployed for use in real life. The high cost of implementing these kinds of solutions does not benefit some small institutions that do not have enough money to buy these systems, yet they also need to secure themselves. This is where Fals-Ism shows its potential as very simple and easy to implement and does not require a lot of computing resources for deployment. Gunawan *et al*. (2021) exploited blockchain technology to verify the authenticity of academic degrees and certificates. However, it is not possible to localize where falsifications appear as proofs.

We implement and analyze the method in (Laptev *et al*., 2017) and compare results to Fals-Ism. Note that the results in (Laptev *et al*., 2017) are discussed using a dataset of 9 PDF documents each of which has only one page. The performance investigation of the method proposed in (Laptev *et al*., 2017) with our datasets provide a robust analysis when compared to the nine other PDF documents that are initially tested in (Laptev *et al*., 2017). We observe the following:

- Laptev *et al*. (2017) method is effective in detecting traces of manipulation made against an image in a document, unlike Fals-Ism, which only detects if the image has been deleted
- Another strength of Laptev *et al*. (2017) is that it can determine the specific software that was used to edit the PDF document
- Laptev *et al*. (2017) require knowledge of some information about the PDF beforehand, unlike Fals-Ism
- Laptev *et al*. (2017) rely on only forgeries using the most common editing software such as adobe photoshop or PDF creator. Fals-Ism operates independently on the PDF editing software

Conclusion and Perspectives

The aim of this study was to propose a system to detect falsified PDF documents, based on graph isomorphism. A general study to understand the structure of PDF documents by analyzing their anatomy was carried out. The extraction of the features of a PDF document allowed us to extract parameters such as structure, content, and metadata from the PDF document. We then translate the extracted features into a graph then an isomorphism check is performed on each graph in order to verify if the PDF is falsified. Experiments were carried out on 36 PDF samples dataset. A comparison of the proposed method was carried out with similar work in the literature and results show that the proposed method is effective and resistant against attacks on the insertion, deletion, and modification of information in the content, structure, and metadata of PDF documents.

Fals-Ism is performed in three levels to look for falsified items. This solution is costly in terms of execution time. The future investigation will be to meticulously optimize its complexity. This aim could be achieved by relying on machine learning automation.

Acknowledgment

The authors greatly appreciate the thoughtful comments made by the manuscript's editors and reviewers.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author's Contributions

Josue Nguinabe: Conception and designed, acquisition of data, Analysis, and Interpretation of results.

Franklin Tchakounte: Conception and designed experiments and analysis of results proofreaded and drafted the article.

Patient Murhula Buhendwa: Mathematical modeling of the system.

Marcellin Atemkeng: Mathematical modeling and analysis of results, proofreaded, and drafted of the article.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Abed, R. M. (2015). Scanned documents forgery detection based on source scanner identification. *American Journal of Information Science and Computer Engineering*, *1*(3), 113-116. http://creativecommons.org/licenses/by-nc/4.0/

Abramova, S. (2016). Detecting copy move forgeries in scanned text documents. *Electronic Imaging*, *2016*(8), 1-9. https://doi.org/10.2352/ISSN.2470-1173.2016.8.MWSF-068

Ali, M. A., & Bhaya, W. S. (2021, May). Higher Education's Certificates Model based on Blockchain Technology. In *Journal of Physics: Conference Series*, (Vol. *1879*, No. 2, p. 022091). IOP Publishing. https://doi.org/10.1088/1742-6596/1879/2/022091

Elkasrawi, S., & Shafait, F. (2014, April). Printer identification using supervised learning for document forgery detection. In *2014 11th IAPR International Workshop on Document Analysis Systems*, (pp. 146-150). IEEE. https://ieeexplore.ieee.org/abstract/document/6830987

Ayinala, H. K., & Grandhi, S. (2021). Text classification from PDF documents. *International Research Journal of Modernization in Engineering Technology and Science*, *3*, 58-63. e-ISSN: 2582-5208

Beaugnon, A., & Husson, A. (2017, June). Le machine learning confronté aux contraintes opérationnelles des systemes de détection. In *SSTIC 2017: Symposium Sur la Sécurité Des Technologies de L'information et des Communications*, (pp. 317-346). https://hal.science/hal-01636303/

Bertrand, R., Gomez-Krämer, P., Terrades, O. R., Franco, P., & Ogier, J. M. (2013, August). A system based on intrinsic features for fraudulent document detection. In *2013 12th International Conference on Document Analysis and Recognition*, (pp. 106-110). IEEE. https://doi.org/10.1109/ICDAR.2013.29

Bertrand, R., Terrades, O. R., Gomez-Krämer, P., Franco, P., & Ogier, J. M. (2015, August). A conditional random field model for font forgery detection. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, (pp. 576-580). IEEE. https://doi.org/10.1109/ICDAR.2015.7333827

Bradley, T. (2011). PDF Files Most Trusted...and Most Targeted. PCWorld. https://www.pcworld.com/article/495492/pdf_files_most_trusted_and_most_targeted.html

Cheddad, A., Condell, J., Curran, K., & McKevitt, P. (2008, November). Combating digital document forgery using new secure information hiding algorithm. In *2008 3rd International Conference on Digital Information Management*, (pp. 922-924). IEEE. https://doi.org/10.1109/ICDIM.2008.4746807

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (1998). *Subgraph Transformations for the Inexact Matching of Attributed Relational Graphs*, (pp. 43-52). Springer Vienna. https://doi.org/10.1007/978-3-7091-6487-7_5

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (1999, September). Performance evaluation of the VF graph matching algorithm. In *Proceedings 10th International Conference on Image Analysis and Processing*, (pp. 1172-1177). IEEE. https://doi.org/10.1109/ICIAP.1999.797762

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2001, May). An improved algorithm for matching large graphs. In *3rd IAPR-TC15 Workshop on Graph-Based Representations in Pattern Recognition*, (pp. 149-159).

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(10), 1367-1372. https://doi.org/10.1109/TPAMI.2004.75

Elhadad, M. K., Li, K. F., & Gebali, F. (2019, August). Fake news detection on social media: A systematic survey. In *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, (pp. 1-8). IEEE. https://doi.org/10.1109/PACRIM47961.2019.8985062

Gebhardt, J., Goldstein, M., Shafait, F., & Dengel, A. (2013, August). Document authentication using printing technique features and unsupervised anomaly detection. In *2013 12th International Conference on Document Analysis and Recognition*, (pp. 479-483). IEEE. https://doi.org/10.1109/ICDAR.2013.102

Gunawan, I. K., Sukmana, H. T., & Ardianto, A. Y. (2021). Blockchain Technology as a Media for Sharing Information that Generates User Access Rights and Incentives. *Blockchain Frontier Technology*, *1*(01), 44-55. https://journal.pandawan.id/b-front/article/view/2

Han, J., Kim, H., Eom, H., & Son, Y. (2021, March). A decentralized document management system using blockchain and secret sharing. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing,* (pp. 305-308). https://doi.org/10.1145/3412841.3442077

Ibrahim, S., Afrakhteh, M., & Salleh, M. (2010, November). Adaptive watermarking for printed document authentication. In *5th International Conference on Computer Sciences and Convergence Information Technology,* (pp. 611-614). IEEE. https://doi.org/10.1109/ICCIT.2010.5711127

Jung, D., Kim, M., & Cho, Y. S. (2022). Detecting Documents with Inconsistent Context. *IEEE Access*, *10*, 98970-98980. https://doi.org/10.1109/ACCESS.2022.3204151

Kada, O., Kurtz, C., van Kieu, C., & Vincent, N. (2022, June). Hologram Detection for Identity Document Authentication. In *Pattern Recognition and Artificial Intelligence: 3rd International Conference, ICPRAI 2022, Paris, France, June 1-3, 2022, Proceedings, Part I* (pp. 346-357). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-09037-0_29

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications*, *80*(8), 11765-11788. https://doi.org/10.1007/s11042-020-10183-2

Khan, M. J., Yousaf, A., Abbas, A., & Khurshid, K. (2018). Deep learning for automated forgery detection in hyperspectral document images. *Journal of Electronic Imaging*, *27*(5), 053001-053001. https://doi.org/10.1117/1.JEI.27.5.053001

Khanna, N., Chiu, G. T., Allebach, J. P., & Delp, E. J. (2008, March). Scanner identification with extension to forgery detection. In *Security, Forensics, Steganography and Watermarking of Multimedia Contents X* (Vol. *6819*, pp. 178-187). SPIE. https://doi.org/10.1117/12.772048

Laptev, P., Ivask, I., & Matulevičius, R. (2017). Method for Effective PDF Files Manipulation Detection. University of Tartu. https://core.ac.uk/download/pdf/237084653.pdf

Patgar, S. V., Rani, K., & Vasudev, T. (2014, November). An unsupervised intelligent system to detect fabrication in photocopy document using Variations in Bounding Box Features. In *2014 International Conference on Contemporary Computing and Informatics (IC3I),* (pp. 670-675). IEEE. https://doi.org/10.1109/IC3I.2014.7019814

Patgar, S. V., & Vasudev, T. (2013). An unsupervised intelligent system to detect fabrication in photocopy document using geometric moments and gray level co-occurrence matrix. *International Journal of Computer Applications*, *74*(12). https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=14fb95dd318b285c1c41f105367c539b63741cf3

Patil, G., Shivakumara, P., Gornale, S. S., Pal, U., & Blumenstein, M. (2022). A new robust approach for altered handwritten text detection. *Multimedia Tools and Applications*, 1-25. https://doi.org/10.1007/s11042-022-14242-8

Perry, B., Carr, S., & Patterson, P. (2000, April). Digital watermarks as a security feature for identity documents. In *Optical Security and Counterfeit Deterrence Techniques III*, (Vol. *3973*, pp. 80-87). SPIE. https://doi.org/10.1117/12.382214

Picard, J., Vielhauer, C., & Thorwirth, N. (2004, June). Towards fraud-proof id documents using multiple data hiding technologies and biometrics. In *Security, Steganography, and Watermarking of Multimedia Contents VI*, (Vol. *5306*, pp. 416-427). SPIE.

PWC. (2022). PWC's global economic crime and fraud survey 2022. https://www.pwc.com/gx/en/forensics/gecsm-2022/PwC-Global-Economic-Crime-and-Fraud-Survey-2022.pdf

Rivera, K., Rohn, C., Donker, J., & Butter, C. (2020). Fighting fraud: A never-ending battle. *PwC's Global Economic Crime and Fraud Survey*. Global Economic Crime Survey.

Schouten, B., & Jacobs, B. (2009). Biometrics and their use in E-passports. *Image and Vision Computing*, *27*(3), 305-312. https://doi.org/10.1016/j.imavis.2008.05.008

Serranito, D., Vasconcelos, A., Guerreiro, S., & Correia, M. (2020, September). Blockchain ecosystem for verifiable qualifications. In *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services, (BRAINS)* (pp. 192-199). IEEE. https://doi.org/10.1109/BRAINS49436.2020.9223305

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22-36. https://doi.org/10.1145/3137597.3137600

Tchakounté, F., Faissal, A., Atemkeng, M., & Ntyam, A. (2020a). A reliable weighting scheme for the aggregation of crowd intelligence to detect fake news. *Information*, *11*(6), 319. https://doi.org/10.3390/info11060319

Tchakounté, F., Kamdem, P. C., Kamgang, J. C., Tchapgnouo, H. B., & Atemkeng, M. (2020b). An Efficient DCT-SVD Steganographic Approach Applied to JPEG Images. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, *7*(24), e2-e2. https://doi.org/10.4108/eai.28-9-2020.166365

Tchakounte, F., Nyassi, V. S., Danga, D. E. H., Udagepola, K. P., & Atemkeng, M. (2021). A game theoretical model for anticipating email spear phishing strategies. *EAI Endorsed Transactions on Scalable Information Systems*, *8*(30), e5-e5. https://doi.org/10.4108/eai.26-5-2020.166354

Van Beusekom, J., Shafait, F., & Breuel, T. M. (2013). Text-line examination for document forgery detection. *International Journal on Document Analysis and Recognition (IJDAR)*, *16*, 189-207. https://doi.org/10.1007/s10032-011-0181-5

Wang, H., He, H., & Zhang, W. (2018). Demadroid: Object reference graph-based malware detection in Android. *Security and Communication Networks*, *2018*. https://doi.org/10.1155/2018/7064131

Yang, C. (2014, June). Fingerprint biometrics for ID document verification. In *2014 9th IEEE Conference on Industrial Electronics and Applications*, (pp. 1441-1445). IEEE. https://doi.org10.1109/ICIEA.2014.6931395

Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., & Liu, H. (2019, July). Unsupervised fake news detection on social media: A generative approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, (Vol. *33*, No. 01, pp. 5644-5651). https://doi.org/10.1609/aaai.v33i01.33015644