American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Multi-Format Document Verification System

Madura Rajapashe[a*], Muammar Adnan[b], Ashen Dissanayaka[c], Dasith Guneratne[d], Kavinga Abeywardane[e]

[a]*Student, Sambodhi Road, Lunuwila 61150, Sri Lanka*

[b]*Student, 3A 115, Allen Avenue, Dehiwala 10350, Sri Lanka*

[c]*Student, 71 hospital road, Nuwara Eliya, , Sri Lanka*

[d]*Student, Ruwan Stores, Uda-Karawita,, Ratnapura 70044, Sri Lanka*

[e]*Lecturer, SLIIT Malabe Campus, New Kandy Rd, Malabe 10115, Sri Lanka*

[a]*Email: it16014046@my.sliit.lk*

[b]*Email: it17086912@my.sliit.lk*

[c]*Email: it17011662@my.sliit.lk*

[d]*Email: it17003292@my.sliit.lk*

[e]*Email: kavinga.y@sliit.lk*

Abstract

The spread of fake documents claiming to be from official sources on social media has led to increasing levels of skepticism and uncertainty in modern society. Currently, there is no easy access method of verification for documents that can be adopted by the public. This paper proposes a method of a multi-format document verification scheme using digital signatures and blockchain. We employ digital signature algorithms to sign document contents extracted using Optical Character Recognition (OCR) methods and attach this signature to the document by converting it into a 2D barcode format. This code can then be used on a shared document to retrieve the document's digital signature and OCR can be used to verify the signature. In addition to this, we also provide an alternative method of verification in the form of forgery detection techniques. These signed documents are stored in a decentralized storage solution backed by blockchain technology, increasing the solution's overall reliability and security.

Keywords: Digital signatures; content extraction; image processing; blockchain; decentralized storage; forgery detection; 2d barcodes.

---

* Corresponding author.

1. Introduction

The digitization of official documents has quickly become the norm, with government and private organizations adopting the digital domain as one of their leading platforms for sharing information with the public. Nowadays, the authenticity of digital content is a significant challenge for both government and the private sector. The falsification or forgery of official documents has become common on the internet, which can be attributed to the wide availability of image manipulation software and the ease of access to digital document forgery techniques[1]. These forged documents usually claim to be noticed from government offices, political parties, and other institutions. With advanced forgery techniques, the documents spread can be made to look identical to real documents. This makes it difficult for the public to verify such documents independently, and even using a fact-checking service may be time-consuming. Apart from that users must rely on the accuracy and independence of the fact-checking service. Research has shown that fake documents usually spread much faster than real or verifiable news[2]. This has led to a significant need for document verification, which is fast, trustworthy, reliable, and secure. This paper aims to provide a document verification solution using digital signatures, 2d barcodes, decentralized storage, blockchain, forgery detection, and content extraction. Using these technologies, the proposed system can be generated signed documents that can be verified regardless of the formats. Content extraction methods are used to extract the contents of a document, and these contents are signed using secure digital signature algorithms, and this signature is represented on the document itself to aid automated verification. The solution ensures the authenticity and integrity of the documents' contents with digital signatures, and decentralized storage and blockchain ensure that documents signed using our method are stored securely. In addition to this, our proposed method provides forgery detection functionality, ensuring forged documents can be accurately identified.

2. Background Study & Related Work

2.1. Digital Signature

Digital Signatures are a method of verifying the authenticity of digital documents by employing cryptographic hashing and asymmetric cryptography[3]. A cryptographic hashing algorithm generates a fixed-length hash for a given message. In the digital signature algorithm, the document to be signed is input to a cryptographic hashing algorithm, and the resulting hash encrypted using the private key of the signer. This produces the digital signature for the document. In the verification process, the digital signature received is decrypted using the public key of the signer, producing the hash of the original signed document. This hash is then compared with the hash generated for the received document. If these hashes match, the document is considered valid and signed by the original signer's private key. This ensures that the digital signature provides both authentication and nonrepudiation. The above-discussed process can be seen in Figure 1.

As a digital signature is only valid for the exact original document that was signed, it quickly becomes inefficient and rigid as it is common practice to convert documents to different formats when sharing them on social media and other platforms. Since social media is becoming one of the main sources of news for people, the spread of misinformation and 'fake news' has become prevalent.

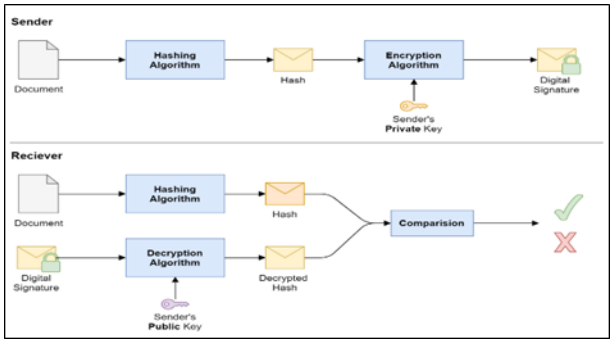

Figure 1: Signature Generation Process

2.2. Decentralized Database & Blockchain

A decentralized database is a database in which multiple databases are distributed over a network in different with no central authority managing the database. A decentralized database can provide advantages over a centralized database in terms of reliability, performance, transparency, and scalability. These advantages make a decentralized database an ideal candidate for storing important documents created by organizations[4]. To enhance the integrity and non-repudiation of the database, blockchain technology can be implemented to ensure that the contents of the database cannot be modified or erased. Using a blockchain can also help the public verify documents created by multiple organizations in one place, instead of having to use a separate solution for each individual organization. To achieve this, we have tested existing decentralized storage solutions like IPFS, Filecoin, and compared each of them in terms of reliability and security. So as the decentralized database, we have chosen IPFS as the most suitable, reliable, and secured database. Blockchain is a system of recording information in a way that makes it difficult or impossible to change. Ethereum can be considered as an open-source platform that uses blockchain technology to construct and execute decentralized digital applications that enable users to directly do the transaction without third party interventions[5]. For the blockchain solution, we have tested Ethereum, Hyperledger, and Multichain and chosen Ethereum as the most suitable blockchain solution which can be integrated with a decentralized database. Ethereum works by means of a worldwide system of PCs that cooperate as a supercomputer. To work with this Ethereum smart contract users need Ether to communicate. Ether is used to secure the blockchain. Ethereum nodes store the most recent state of each smart contract and this is in addition to all the other transactions. Monitoring the current state of the Ethereum application which includes the user's balance and smart contract code with its stored location is a must for each Ethereum application. Ethereum transactions are similar to bank account transactions as it is maintained under a particular account number. While blockchains are secure and can be used to store data, it is not suitable for storing large amounts of data such as PDF documents or images. This is because the data must be duplicated across a chain of thousands of machines and this process can be very expensive. This illustrates that hundreds and thousands of computers working together so they can verify data by reaching a "consensus". The consensus illustrates that the uploaded data is valid. There is a limitation for the amount of data that can be processed at a given block. In Ethereum it's 15 seconds per block and when it comes to the average amount of data that can put into a new block is around 20KB. Ethereum is so powerful because of its consensus process. Signed PDFs or Documents and Security video footages or photos are a few examples where timestamped and tamperproof data can be useful. So, this research focuses on Signed PDFs or documents and Images[6]. As decentralized storage

uses hashes to identify documents, the storage solution can be used with a blockchain to ensure immutability. The consensus process of Ethereum assures that the added data is immutable, so it is a good way to store hashes of the documents[7].

2.3. Content Extraction & Image Processing

Content extraction can be used to retrieve text and other contents from any document format such as PDF, JPEG, PNG, etc. These extracted contents are in a machine-readable format and can be processed or stored in a database. Some commonly used methods include Connected Component-Based, Edge Based, Region-Based, Mathematical Morphology Based, and Digital Image Processing methods[8]. These methods utilize various techniques to detect and identify characters and retrieve text from images. The accuracy of these techniques depends greatly on the quality of the input image; therefore, image processing methods can be employed to better prepare the images for content extraction. Some of these image processing methods include: 2D Convolution, Averaging Algorithm, Gaussian Algorithm, Median Filtering Algorithm, and Bilateral Filtering Algorithm

2.4. Forgery Detection

Digital document forgery detection is a complex task; however, it is possible to achieve a reasonable success rate using appropriate methods. These include verifying the consistency of the text lines, analyzing the headers of the document, analyzing the variations in the alignments, and the rotations of the added text-lines compared to previously issued authentic documents. Furthermore, techniques such as fuzzy hashing and perceptual hashing can be employed to detect forgeries. These methods can be used to compare the differences between two documents based on their features. While they employ hashing methods, these algorithms can ignore small changes that can occur due to file conversion and quality loss[9,10,11,12].

2.5. Previous Studies

In the a previous work[13] the authors propose using wave atom transform based hashing alogrithms to verify the authenticity of printed identity documetns such as passports, national identity crds, etc. In this paper a WAT-based hash is signed using a digital signature and encoded to a QR code which is attached to the document and can be used for verification. In another multi-format document verification research[14] the authors propose signing the message in the document, and compressing and encoding the generated signature into a QR which can be attached to the document. In this research the authors propose using OCR to verify the signature retrieved from the QR code. In the previous works[15] one of the author propsed a solution which contains a verification ID that can be used for blockchain-based authenticity of the digital assets. This ID is instinctively a block that can be used for verifying the e-documents. Another solution was to track the authenticity of the digital assets. By converting the digital contents to the binary file and storing the generated hash for that binary file in the blockchain[15]. This hash works as the identifier for the owner. This system deviates from decentralization and it follows the centralization concept and it has become uncertain for integrity breaches Our solution improves upon existing research in the areas of content extraction, decentralized storage, and forgery detection. Proposed

solution helps to improve the accuracy, performance, and the security of multi-format document authentication.

3. Methodology

In this section, the authors outline the various methods used to implement the proposed solution. Figure 2 is an overview of the overall process conducted in the solution.

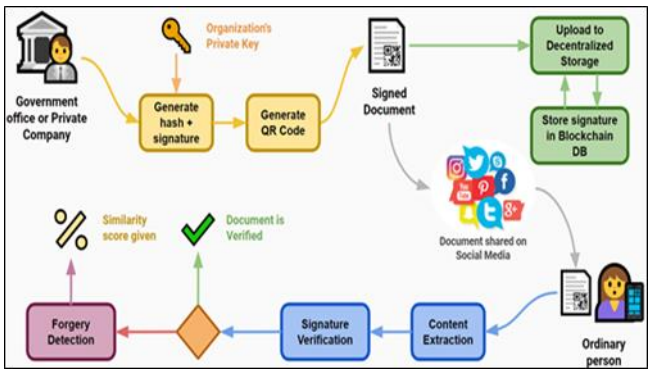

Figure 2: Overall Functionalities

In the overall process, organizations can sign documents using their private key. The signing process uses content extraction to extract the contents from the document, and these contents are signed using a digital signature algorithm. This signature is then converted to a QR code and attached to the document. This document is then uploaded to the decentralized storage, and the hash of the document and related metadata are stored on the blockchain. After this, the document can be converted to any format and shared on social media. Once a user uploads the document to verify it, content extraction is used to extract the contents of the document and signature metadata from the uploaded documents. These contents and signature are cross-checked with the blockchain to ensure it is valid, and the document is verified using the organization's public key. If the document is invalid, forgery detection can be used to give the user an idea of the validity of the document. In the following sections, these components will be explained further.

3.1. Signature Generation

In this process, a digital signature is produced for the contents of the document, and this signature is converted to a QR (Quick Response) code along with other metadata relating to the document. The document's contents are first extracted using content extraction, and these contents are hashed and signed using SHA-256 and ECDSA algorithms respectively. For the signing function, a prime-256 Elliptic Curve Cryptography belonging to the signer is used. This generated signature is then added to the document metadata object. This object is structured as follows: Metadata version, Organization ID, Document ID, Hashing and Digital Signature algorithm, and Digital Signature of the document's contents. The digital signature in this metadata is encoded to base64 to ensure compatibility with the QR code generation process[16,17,14]. Once the metadata is formatted, it is then converted to a string where all the fields are separated by commas. This metadata is then converted to a QR code which can then be attached back to the document. This document can then be shared and stored in decentralized storage. A sample result of scanning a QR code can be seen in Figure 3.

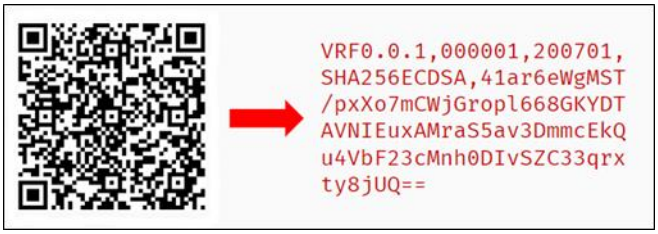

Figure 3: QR Image

This QR code can then be attached to the original document to create the signed document. This signed document is then uploaded to the decentralized storage.

3.2. Decentralized Storage & Blockchain

In this process, the signed document is uploaded to the Interplanetary File System (IPFS). After adding data to the IPFS, it will return a hash for the added content. The same hash will be return if the content is the same. The reverse process is content retrieval from IPFS. Throughout this retrieval process, the requester of the content cryptographically guaranteed to receive the same content that was initially uploaded to the IPFS network. This process is more powerful if it combines with Ethereum. IPFS is the solution for storage issues in the blockchain and by that, we can extend the limitation in the blockchain[7]. So, if we want to store a large piece of data, we can upload the image or document to the IPFS and the returned hash can be uploaded to the Ethereum. Before any organization uses VERIFI, first they need to register for the application. Organization registration details are added to the Ethereum by the application. After uploading the file to the VERIFI it will automatically be shared with the IPFS and the hash will be sent to the Ethereum. In the complete implementation process, there are two main phases. Those are File creation and Storage and File Retrieval. After uploading an image to the VERIFI application, VERIFI provides a public key and a private key to each organization. VERIFI generates private keys for the signing process and public keys to verify the process. After uploading IPFS returns the uploaded hash as shown below[7]. Refer the Figure 4 for a detailed description. In this research project, we have customized the Ethereum smart contract to store metadata in the blockchain. So, this smart includes metadata which includes signature, content Hash, document Hash, Organization ID, and Document ID. VERIFI application will receive the deployed contract address of the smart contract after the deployment.

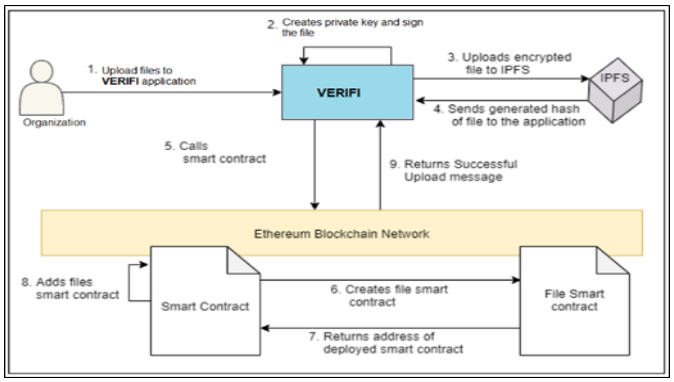

Figure 4: IPFS Functionality

3.3. Image Processing & Content Extraction

This process is used to extract contents from documents or images and be integrated with signature generation and verification components. Image text can be found in captured images, scanned documents, magazines, newspapers, posters, and notices, etc. Most of those images are noisy. Therefore, if a user or an organization uploads an image to the system and if the image is noisy, prior to the content extraction, noisy and low-quality images need to be cleaned up. For that purpose, "OpenCV" which is an open-source image processing and machine learning library is used to clean up and to obtain higher quality images[8]. Median Filter is a nonlinear digital filtering technique, often used to remove noise from an image. It is based on an algorithm called "Median Filtering Algorithm". The median filter is the best noise-reducing filter so far. The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The difference between an original image and a filtered image can be seen in Figure 5. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For one-dimensional signals, the most obvious window is just the first few preceding and following entries, whereas for two-dimensional (or higher-dimensional) data the window must include all entries within a given radius or ellipsoidal region.

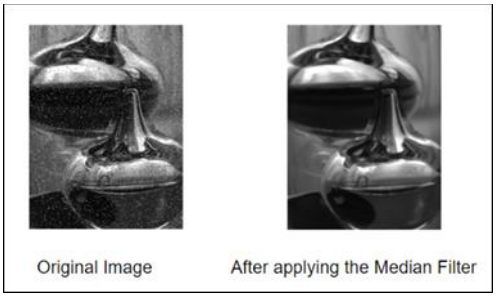

Figure 5: Median Filter

After the cleaning up process is done, images or PDFs are delivered for the content extraction process. For that authors utilizes "Tesseract OCR" (Optical Character Recognition). Tesseract is an OCR engine with support for Unicode and the ability to recognize multiple languages (100 languages). It can also be trained to recognize other languages. This process ensures that documents of any format can be verified using our method. This component is used in both the signature generation, to extract the documents contents for signing, and the signature verification component, to extract the document contents and signature metadata from the QR code, for signature verification.

3.4. Signature Verification

In this process, the contents from the received document are extracted using content extraction to retrieve the document text and the QR code. The QR code contains the document metadata which can be used to retrieve the public key of the organization based on the organization ID. A hash can then be generated for the extracted text contents and this hash. Consequently, a hash can be input to the signature verification function along with the public key of the signer and the received digital signature. This function will output a result of either true or false indicating whether the document is valid or not.

3.5. Forgery Detection

In this section, the primary purpose is to detect any forgeries that have been done to documents. Proposed forgery detection implementation is more sophisticated than most prevailing methods as the process is built as a combination of multiple forgery detection techniques. The techniques are text similarity using cosine algorithms, document similarity checking, the perceptual algorithm for detecting image similarities, and image comparison using pixel comparison. After obtaining results from these algorithms, the Forgery detection function can predict the similarity percentage of a particular document, news article, notices using the proposed machine learning algorithm. The cosine algorithm measures the similarity between two nonzero vectors by measuring the cosine angle between nonzero vectors. The algorithm begins with the tokenization of words. The tokenization has two parts, which are the process of dividing significant sentences into small pieces called tokens, and the next part is getting a list of stop words such as 'a', 'in', 'the'. The next step is to get the tokenized string sets and remove the stop words from both sets. Then it is possible to create vectors using word sets without any stop words. The vectors (A, B) and union of the same vectors (||A||, ||B||) go through an AND operation, and by dividing those vector sets, it is possible to get the similarity ($Cos\theta$) between two vectors as a percentage as seen in(1)[9]

$$\text{Cosq} = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|\|\vec{B}\|} = \frac{\sum_i^n a_i b_i}{\sqrt{\sum_i^n a_i^2}\sqrt{\sum_i^n b_i^2}} \qquad (1)$$

The logic of the document similarity is the same as the cosine text similarity algorithm. First, the algorithm opens the documents and reads them. Then as in the cosine similarity, the document similarity checking algorithm gets the line numbers from both of the documents. The text in the documents goes through a tokenization process. The tokenized word lists help to identify the word frequency of both the documents. From the tokenization, the algorithm can give an output of line numbers, word count, and the number of distinct words in two documents. After determining those outputs, the algorithm is capable of giving an output of the difference between two documents by putting the word sets through the dot production algorithm, which is the same as the cosine algorithm. The next algorithm is the perceptual algorithm to check similarities between images. The perceptual hashing algorithm has four sub algorithms which are the average hash, difference hash, perception hash, and wavelet hash. The average hash creates 64 bits hashes by using the properties of the image. The image is divided into components that have a higher frequency and lower frequency. Higher frequency is responsible for image details, and lower frequency is responsible for image structure. The average hash scales down the image into 8 x 8 blocks which contain 64 pixels and those pixels convert into grayscale. Then the algorithm calculates an average color to all the pixels. Those pixels are used to create the hash specific to the image[10]. The difference hashing algorithm is based on the structure of the image and it divides the image into 7 x 9 blocks and converts those blocks into grayscale. Then in each row computing the difference between two pixels. Altogether there are 64 differences computed using this method. Then this algorithm creates a fingerprint using the above data. After getting the hashes from images, it is possible to use the hamming distance method to get the difference from the two hashes. The next sub-algorithm is perception hash. This sub-algorithm gets the image and converts it into grayscale using adding a filter to the 7x7 dimension kernel. Then the filtered image resizes into 32 x 32 pixels to extract the hash and This hashing algorithm is based on Discrete Cosine

Transformation (DCT). Wavelet hashing converts the original image into the frequency domain to create the hash and wavelet hashing algorithm based on the Wavelet Discrete Transform method[10]. This algorithm is predicting the similarity between two images using image pixels. First, the algorithm reads the two images and checks for size comparison and RGB channels of the two images. Then it is possible to identify any different pixels in two images. The next step is to create key points. Scale-Invariant Feature Transform (SIFT) method used to find key points and SIFT can detect high relevant features and determine the relevance degree called descriptors[12]. This description can be used to match key points in different images. Using those key points on two images, matching points in two images can be obtained. Consequently, using matching points and key points, it is possible to determine the similarity percentage of two images. The result is predicted using a machine learning model. The machine learning model is created using the logistic regression algorithm. To this model, the logistic regression has dichotomous exposure (e) and a single dichotomous cofounder (z). The model and the results obtained from this algorithm can enhance the use of multiple categorical and continuous cofounders. The following algorithm can be used to get the outcome ($\hat{P}_{ez}$) as a probability using logistic regression. Following algorithms $\hat{\alpha}$, $\hat{\beta}_1, \hat{\beta}_2$ are considered as estimated regression coefficients, as seen in (2)[11].

$$\hat{P}_{ez} = \frac{\exp[\hat{a} + \hat{b}_1 \times e + \hat{b}_2 \times z]}{(1 + \exp[\hat{a} + \hat{b}_1 \times e + \hat{b}_2 \times z])} \quad (2)$$

The outputs of forgery detection algorithms mentioned above are used as the input to the machine learning model dataset. Based on the data fed into the model, it is possible to get the probability of possible forgeries that have been done to a particular document, article, or notice.

4. Results and Discussions

To implement the Signature Generation component, multiple hashing algorithms, signature algorithms, and 2d barcode formats were tested. The results of testing QR codes can be seen in Figure 6. The chosen algorithms considered the output sizes and speed. SHA-256 and Elliptic Curve Digital Signature Algorithm or ECDSA were chosen for a signature generation as they are the recommended algorithms for digital signatures by the NIST. During our testing process we found that signature generation with ECDSA generally takes less than 1 millisecond, while producing an output size of 512 bits. While this is a relatively small output size, a Prime-256 ECC key provides the level of security as a 3078-bit RSA key [18]. In addition to this, the small output size ensures the generated barcode will fit on the page of the signed document without obstructing content while still being scannable. QR code was chosen as the barcode format as it can contain 2953 bytes of data while being relatively small and providing up to 4 levels of error correction.

Figure 6: QR Types

As the final results in IPFS and Blockchain, A SHA-256 hash has been generated and the particular hash is sent to the blockchain. Under that other details such as Metadata version, Organization ID, Document ID, Hashing and Digital Signature algorithm, and Digital Signature of the document's contents are saved in the blockchain. Since the speed of decentralized storage solutions depend on the number of nodes closest to the user, the upload and retrieval process has good performance, and helps reduce performance bottlenecks that traditional centralized databases face. As the result of "Image processing" and "Content Extraction", the Original uploaded image or PDF goes through our solution and if the images are noisy then it goes through the Median filter and gives a better output after the image cleaning process. After that, the cleaned image goes through the Tesseract OCR and extracts the content successfully as a text file. As OCR greatly depends on the quality of the input image, the cleaning process helps improve the accuracy of the content extraction giving results of up to 99% accuracy. After the content extraction process, the final output of the text file is delivered to the signature generation and verification components. In forgery detection, the Cosine algorithm gives the result as a percentage by comparing two content using a union method. Document similarity algorithm output is the number of words, lines, and distinct words. D-Hash algorithm compares the two hashes from two images and gives an output of difference using hamming distance, and the final algorithm is the image similarity algorithm which gives the percentage of similar pixels. All four outputs are used to train the machine learning model and predict the result using the logistic regression algorithm as a binary value. The Machine Learning Model has 94% accuracy according to the current dataset that is being used in the research component. The system supports many cases like false negatives. So, this can be illustrated through the following outputs of the VERIFI. This result shows that a valid document which has been successfully verified using the system. the following result illustrates the successful verification of the signature of the document using digital signature verification. Refer the Figure 7.

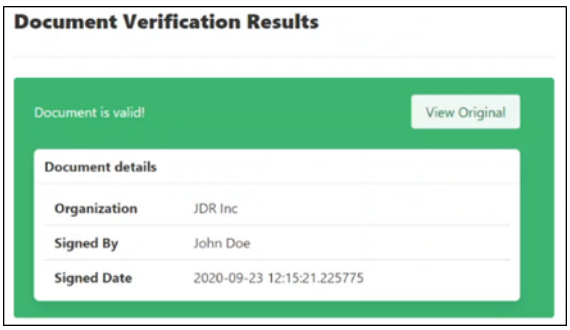

Figure 7: Successfull Verification

The following result shows a low-quality valid document that failed to be verified. In this case, the uploaded document is valid, but content extraction may have failed due to the quality of the image, making the signature invalid. This is a false negative case. The similarity results can give an idea to the user whether the document is valid or not. A content similarity result above 95% ensures the user that the contents of the documents are very similar. Refer the Figure 8.

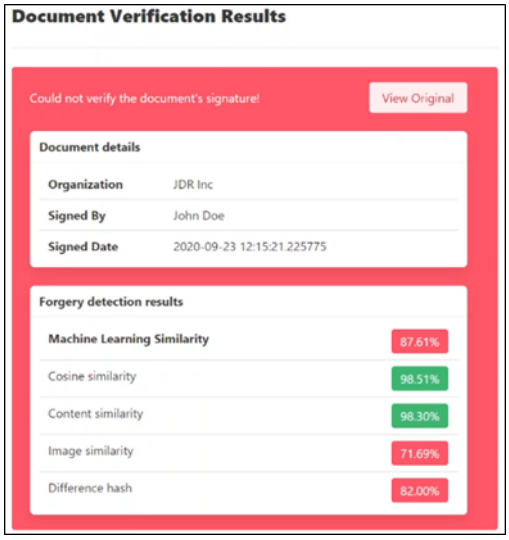

Figure 8: Unsuccessfull Verification 1

The third result shows a forged document being successfully identified, and the similarity results being shown to the user. In this case, a false document has been created with a valid QR code attached. Here users will be able to identify that the forgery detection results are very low, indicating a forged document. For further verification, the user is able to view the original signed document that relates to the attached QR code. Refer the Figure 9.

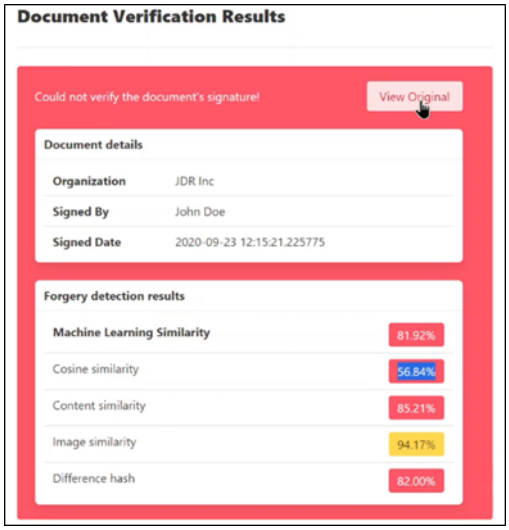

Figure 9: Unsuccessfull Verification 2

As a result, we can state that VERIFI is a working system that can sign documents uniquely and store documents in IPFS, store hash on Ethereum, and VERIFI can identify forged documents using Machine Learning

5. Conclusion

In conclusion, we believe our the outcome of our research will provide an easy way to use and secure method of document verification regardless of the format of the document. With the evolution of the Internet and the

increase of the spread of fake news on social media websites, this solution will help users verify the authenticity of documents shared through unverified sources. While there are existing research that addresses the problem of multi-format document verification, they do not provide fully automated verification. Our research improves upon the lacking features and adds unique features such as a decentralized storage solution and improved forgery detection features. We believe these features improve its accuracy and reliability. We believe this will help both public and private organizations share their notices and other documents to the public without worry of having their documents forged. Furthermore, we hope it will help the public verify documents shared on social media claiming to be from official sources.

6. Recommendations

This work can be further extended to include support for documents in languages other than English, support for considering the images included in the documents, and more. With these changes, this work can be used in more applications such as verification of identity cards, passports, and many more. Currently the solution only supports single-page documents, and further research can be conducted in a method to verify multi-page documents. Furthermore, more research can be conducted in the area of content extraction to improve the accuracy and relaibility of the solution. In addition to this, the forgery detection component also has room for improvement with the evolution of machine learning.

Acknowledgements

We appreciate the guidance, support, and encouragement given by our supervisor and examiners. We would also like to extend our gratitude to the Sri Lanka Institute of Information Technology for providing us with a platform to complete this research successfully.

References

[1]. S. Nelatury, “Digital forgery,” no. May 2017, 2018.

[2]. S. Vosoughi, D. Roy, and S. Aral, “News On-line,” Science (80-.)., vol. 1151, no. March, pp. 1146–1151, 2018.

[3]. W. DIEKE, “Die Klinik der Seifenaborte. Bemerkungen zur Arbeit von Dr. Horst Scholz,” Arztl. Wochensch., vol. 7, no. 26, pp. 611–613, 1952.

[4]. K. K. Ezéchiel, S. Kant, and R. Agarwal, “A systematic review on distributed databases systems and their techniques,” J. Theor. Appl. Inf. Technol., vol. 97, no. 1, pp. 236–266, 2019.

[5]. M. Xu, X. Chen, and G. Kou, “A systematic review of blockchain,” Financ. Innov., vol. 5, no. 1, 2019.

[6]. A. Tenorio-Fornés, S. Hassan, and J. Pavón, “Open peer-to-peer systems over blockchain and IPFS: An agent oriented framework,” CRYBLOCK 2018 - Proc. 1st Work. Cryptocurrencies Blockchains Distrib. Syst. Part MobiSys 2018, pp. 19–24, 2018.

[7]. S. Khatal, J. Rane, D. Patel, P. Patel, and Y. Busnel, “FileShare: A Blockchain and IPFS Framework for Secure File Sharing and Data Provenance,” pp. 825–833, 2021.

[8]. A. Nair, “Overview of Tesseract OCR engine An overview of Tesseract OCR Engine Seminar Report

Akhil S B130625CS Department of Computer Science and Engineering National Institute of Technology , Calicut Monsoon-2016," no. December 2016, 2017.

[9]. B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8206 LNCS, pp. 611–618, 2013.

[10]. A. Drmic, M. Silic, G. Delac, K. Vladimir, and A. S. Kurdija, "Evaluating robustness of perceptual image hashing algorithms," 2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc., pp. 995–1000, 2017.

[11]. C. J. Muller and R. F. Maclehose, "Estimating predicted probabilities from logistic regression: Different methods correspond to different target populations," Int. J. Epidemiol., vol. 43, no. 3, pp. 962–970, 2014.

[12]. O. Andersson and S. R. Marquez, "A comparison of object detection algorithms using unmanipulated testing images: Comparing SIFT, KAZE, AKAZE and ORB," 2016.

[13]. F. Ahmad and L. M. Cheng, Paper Document Authentication Using Print-Scan Resistant Image Hashing and Public-Key Cryptography, vol. 11611 LNCS, no. July. Springer International Publishing, 2019.

[14]. M. Warasart and P. Kuacharoen, "Paper-based Document Authentication using Digital Signature and QR Code," 4TH Int. Conf. Comput. Eng. Technol., vol. 40, no. January, pp. 94–98, 2012.

[15]. N. Nizamuddin, H. R. Hasan, and K. Salah, "IPFS-blockchain-based authenticity of online publications," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10974 LNCS, no. June, pp. 199–212, 2018.

[16]. M. Alidoost Nia, A. Sajedi, and A. Jamshidpey, "An Introduction to Digital Signature Schemes," no. April, 2011.

[17]. A. Singhal and R. S. Pavithr, "Degree Certificate Authentication using QR Code and Smartphone," Int. J. Comput. Appl., vol. 120, no. 16, pp. 38–43, 2015.

[18]. Pycryptodome, "ECC." [Online]. Available: https://pycryptodome.readthedocs.io/en/latest/src/public_key/ecc.html.