American Journal of Artificial Intelligence
2026; 10(2): 189-197
http://www.sciencepublishinggroup.com/j/ajai
doi: 10.11648/j.ajai.20261002.12

Verification Challenges of AI-Generated Identity Documents: Blockchain Technology as a Trust Layer and AI as a Supporting Signal

Tapendra Baduwal*

Department of Information Technology, Presidential Graduate School, Kathmandu, Nepal

Email address:
tapendrabaduwal200@gmail.com (Tapendra Baduwal)

To cite this article:
Tapendra Baduwal. (2026). Verification Challenges of AI-Generated Identity Documents: Blockchain Technology as a Trust Layer and AI as a Supporting Signal. *American Journal of Artificial Intelligence, 10*(2), 189-197. https://doi.org/10.11648/j.ajai.20261002.12

Received: 14 May 2026; **Accepted:** 25 May 2026; **Published:** 2 July 2026

Abstract: The rapid advancement of generative AI has revolutionized digitalization, while simultaneously introducing new security challenges. As AI models are increasingly integrated into cameras to enhance or modify images, a fundamental question arises for verification systems: whether captured images retain authentic camera fingerprints, such as Photo-Response Non-Uniformity (PRNU), Color Filter Array (CFA) patterns, physically random sensor noise, and lens distortions, or are heavily altered or fully generated by AI. Modern generative AI models create images that are highly similar to those produced by cameras, increasing the risk of document forgery and verification challenges. To address these challenges, this research proposes blockchain technology as a foundational trust layer for digital identity, enabling secure and tamper-proof evidence recording through an immutable ledger and cryptographic mechanisms. The proposed system integrates blockchain with a layered microservices architecture, separating user management, blockchain interaction, and audit logging into independent services. Communication between services uses gRPC with clearly defined Protocol Buffer schemas for efficient communication. The API layer is implemented using FastAPI for authentication, authorization, and request routing with high performance and automatic documentation. Data is stored in MongoDB, including user profiles, authentication records, verification results, and audit logs, which ensures flexibility and high availability. AI is used as a supporting signal rather than a definitive decision-maker. Experimental evaluation was conducted on 4,550 handwritten signatures, created using real ink pens but not belonging to any specific individual, and 4,550 AI-generated signatures were created using OpenAI's GPT image models, Nano Banana 2, and Qwen image generation models. ResNet50 was used to compute the signal score and achieved an F1 score of 0.996 on the classification task. The proposed method is designed to generalize well across a wide range of document and image domains.

Keywords: Generative AI, Verification Challenges, Digital Identity, Blockchain Technology, Cryptographic Mechanisms

1. Introduction

Recent generative AI models such as OpenAI's GPT image models, Nano Banana 2, and Qwen image generation models are now capable of generating highly realistic images that look very similar to real-world photographs. By learning from large-scale image-text datasets, these models can generate realistic textures, edges, and structural details. When provided with a text prompt or an input image, these models can produce or modify visual content that appears highly natural and convincing [1, 2]. These capabilities introduce significant challenges for AI-generated document verification systems.

Blockchain is a decentralized ledger technology that records transactions across multiple computers, ensuring transparency, security, and immutability. Transactions are grouped into blocks containing details, timestamps, and cryptographic hashes, which are linked sequentially to form a tamper-proof chain. Blockchain validates each transaction through consensus mechanisms, maintaining integrity and trust across the network [3].

Blockchain technology has evolved through major

milestones in cryptography and digital innovation. In 2008, Bitcoin was introduced by an anonymous individual or group known as Satoshi Nakamoto as a decentralized digital currency, enabling peer-to-peer transactions without intermediaries. In 2009, the first functional blockchain went live using a proof-of-work mechanism to securely record transactions and prevent double-spending. In 2013, Vitalik Buterin proposed Ethereum, introducing smart contracts that automatically execute when conditions are met. By 2017, blockchain gained widespread attention with Initial Coin Offerings (ICOs). In the 2020s, it has expanded beyond cryptocurrencies into finance, healthcare, supply chain management, real estate, and voting systems, integrating with technologies like AI and the Internet of Things (IoT) [4–6].

The combination of AI and blockchain ensures data integrity and transparency through blockchain, while AI performs intelligent analysis. Each verification result is securely stored on the blockchain, keeping both original and updated records along with a clear audit trail. Together, they establish a robust and scalable ecosystem for secure digital verification [7].

Traditional identity verification systems rely on centralized databases, which makes them vulnerable to breaches, fraud, and unauthorized access. Blockchain technology provides a decentralized and tamper-resistant framework for managing digital identities. Using Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), users can securely store and control their personal information. Verification requests are processed in real time, with cryptographic proofs ensuring authenticity without exposing sensitive data. This approach enhances privacy, trust, and security across sectors such as finance, healthcare, and government services [8].

The rapid growth of digital services has led to large amounts of personal information being stored online, making it more vulnerable to cyberattacks like identity fraud. Recent advances in AI, including deepfake technology, allow fraudsters to create very realistic fake documents, photos, and videos, making them harder to detect. These developments pose significant challenges for identity verification systems and digital service providers [9].

1.1. Problem Statement

Recent advances in generative AI enable the creation of highly realistic synthetic signatures and identity documents by learning patterns from large-scale datasets, producing outputs that closely mimic natural camera captures as well as genuine handwriting and content. As AI models are increasingly integrated into cameras to enhance or modify images, verification systems face a fundamental challenge: determining whether captured images retain authentic camera fingerprints, such as Photo-Response Non-Uniformity (PRNU), Color Filter Array (CFA) patterns, sensor noise, and lens distortions, or are heavily altered or fully generated by AI. As AI image generation models such as OpenAI's GPT image models, Nano Banana 2, and Qwen models continue to evolve, they increasingly challenge traditional verification systems. These developments increase forgery risks and make it harder to verify AI-generated documents in the era of generative AI [1, 2, 10].

1.2. Contribution

The major contributions of this study are summarized as follows:

1) Research on verification challenges caused by modern generative AI models, proposing a blockchain-based trust layer that uses cryptographic mechanisms and audit trails to ensure tamper-proof digital identity verification.
2) Perform an experimental evaluation using ResNet50 to distinguish images generated by state-of-the-art generative AI models from real camera-captured images and produce confidence scores to estimate the likelihood of an image being AI-generated.
3) Created a dataset of handwritten signatures that do not belong to any specific individual to ensure confidentiality. AI-generated signatures were created using advanced generative AI models. The dataset is publicly available for research use.

Overall, this research explains the verification challenges caused by generative AI models and how blockchain can help build secure and trustworthy verification systems.

2. Literature Review

This study reviews blockchain-based identity verification systems and the challenges introduced by generative AI models in identifying AI-generated documents, and examines AI-based techniques for detecting synthetic documents.

Rahman et al. (2023) proposed a blockchain-based academic certificate authentication system to address certificate fraud, which poses serious risks to institutional integrity across education and employment sectors. While other blockchain certificate systems can only generate and verify certificates leaving errors such as wrong names or dates unfixable after generation, their system introduces a correction mechanism where a new corrected block is created and linked to the original through a dedicated correction chain, preserving full blockchain immutability while enabling traceable error management [11]. However, it is limited to structured certificate data and does not address AI-generated documents. To handle such cases, our approach integrates blockchain with AI-based verification signals to enhance robustness against emerging generative AI forgeries.

A study by Lopes et al. (2025) presented a distributed digital identity management system integrating blockchain simulation with Ethereum and Ganache, demonstrating the practical feasibility of decentralized technologies for secure and auditable identity management. The system adopted a microservices architecture with JWT-based authentication and comprehensive audit logging to ensure traceability and access control, while Ganache simulated blockchain operations to record identity transactions immutably and was validated in a local environment. The authors acknowledged that production

deployment would require integration with a real Ethereum network, highlighting scalability, computational cost, and real-world feasibility as key challenges [12].

Researcher Vinogradov (2025) investigates whether diffusion-based generative models including Stable Diffusion, Qwen, Flux, and Nano-Banana can forge fraudulent identity documents capable of bypassing digital verification systems. Although these models successfully reproduce the general visual appearance of identity documents such as layout and color scheme, they consistently fail to replicate physical security features including laser engraving textures, holograms, and microprinting that forensic professionals depend upon for authenticity assessment. The study identifies a key verification gap, showing that digital platforms relying only on surface-level visual inspection remain vulnerable to imperfect forgeries, especially under low-resolution or compressed image conditions [13].

Hao and Zheng (2025) studied the problem of detecting AI-generated forged handwritten signatures in the context of emerging generative technologies. They constructed synthetic signature datasets using multiple AI-based font generation tools to simulate realistic forgeries. The study utilized a Data-Efficient Image Transformer (DeiT) architecture, which leverages self-attention mechanisms for feature extraction. This approach enables the model to capture both local stroke characteristics and global structural relationships within signature images [14]. However, AI-based verification alone is not enough, as generative AI keeps improving and can produce highly realistic images.

Abdirahma et al. (2024) proposed an offline handwritten signature verification system to distinguish genuine signatures from forgeries. A preprocessing pipeline incorporating resizing, normalization, and data augmentation was applied to eliminate noise and capture essential patterns across diverse signature styles. YOLOv5 was integrated for real-time signature localization and detection within input images, while MobileNet served as the primary classification backbone for efficient feature extraction from the detected regions [15].

Researcher Gupta (2025) conducted a comprehensive review of security risks posed by generative AI in financial systems, identifying deepfakes, synthetic identity fraud, and AI-generated phishing as major threats. Deepfakes can bypass Know Your Customer (KYC) and Anti-Money Laundering (AML) authentication by convincingly replicating identities, enabling unauthorized access and fraud. Synthetic identity fraud mixes real and fake data to evade verification checks, while AI-generated phishing uses natural language processing and behavioral analysis to create highly targeted social engineering attacks. To mitigate these risks, the study recommends multi-factor authentication, liveness detection, adversarial training, continuous monitoring, and strong AI governance [16].

After reviewing various studies, it is found that methods focus either on blockchain-based identity management or AI-based forgery detection. To address this gap, this study proposes a combined blockchain and AI-based verification approach for tamper-proof evidence recording and improved detection of AI-generated forgeries.

3. Blockchain Technology

3.1. Fundamental Components

3.1.1. Distributed Ledger

Every node in the network maintains an identical copy of the ledger, giving all participants equal access to the same immutable chain of transactions. With no central authority holding a master record, the system achieves transparency and fault tolerance through decentralization, making the ledger a trustworthy and tamper-resistant source of truth across the network [17, 21].

3.1.2. Immutable Records

Once a transaction is recorded on the ledger, no participant can alter or tamper with it. Any error requires a new corrective transaction to be appended rather than overwriting the original, meaning both entries remain permanently visible on the chain. This ensures a complete and verifiable audit trail, making the ledger a trustworthy and tamper-resistant source of truth across the network [17, 21].

3.1.3. Smart Contracts

Smart contracts are self-executing programs stored on the blockchain that automatically enforce predefined rules when specified conditions are met. They eliminate the need for intermediaries, reducing delays and human error in processes such as bond transfers, insurance payments, and identity verification [19, 21].

3.1.4. Consensus Mechanism

Blockchain networks use consensus algorithms to validate transactions and ensure a consistent ledger state across all nodes. Rather than depending on a central authority to approve records, all participants collectively reach agreement through a defined protocol. Proof of Work (PoW) requires nodes to solve complex puzzles to validate blocks, while Proof of Stake (PoS) selects validators based on their staked tokens [18, 22].

3.1.5. Cryptographic Security

Each block contains the hash of the previous block, so tampering with one block invalidates the entire chain. Security is reinforced by public key cryptography, where a public key acts as an address and a private key authorizes transactions via digital signatures, ensuring ownership and preventing unauthorized modifications [20, 21].

3.2. Blockchain Operation Process

3.2.1. Records Transactions as Blocks

Each transaction is recorded as a block of data on the blockchain, capturing key details about the transfer of assets. An asset is anything of value that can be owned, transferred, or tracked on the blockchain. Each block includes essential information such as the parties involved, transaction type,

time, location, amount, conditions, and a timestamp [21, 22].

3.2.2. Connects Blocks Together

Each block is connected to the previous block and the next block, forming a secure chain. This is done using cryptographic hashes, which are unique codes for each block. Each hash depends on the data of the previous block, ensuring the correct order of transactions. Because of this, altering any block requires modifying all subsequent blocks, ensuring high security and data integrity [21, 22].

3.2.3. Builds an Irreversible Blockchain

Blocks are linked into an immutable chain, with each new block validating previous ones. Network nodes validate transactions through consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS), ensuring security and consistency across the decentralized system [21, 22].

3.3. Types of Blockchain Networks

3.3.1. Public Blockchain Networks

A public blockchain is an open, decentralized network where anyone can join, read, and validate transactions, but with higher cost, low privacy, and strong transparency and security. Example: Bitcoin, Ethereum [21, 23].

3.3.2. Private Blockchain Networks

A private blockchain is a decentralized peer-to-peer network where a single organization controls who can participate, run consensus, and maintain the shared ledger. Example: A hospital managing patient records accessible only to authorized medical staff [21, 23].

3.3.3. Permissioned Blockchain Networks

Businesses that create private blockchains usually use permissioned networks, though public blockchains can also be permissioned. This restricts who can participate and record transactions, requiring approval to join. Example: A company blockchain accessible only to authorized employees [21, 23].

3.3.4. Consortium Blockchain Networks

A consortium blockchain is jointly managed by multiple trusted organizations that share control of the network. It is used when collaboration between institutions is required. Example: A group of banks sharing a common blockchain system [21, 23].

3.4. Blockchain Protocols and Platforms

Blockchain protocols define the rules for recording, sharing, and securing data in a blockchain network, while platforms provide the infrastructure and tools to build and deploy decentralized applications [21, 23].

3.4.1. Hyperledger Fabric

Hyperledger Fabric is an open-source, modular blockchain platform that provides tools and libraries for building private, enterprise applications, offering identity management and access control for use cases such as supply chains, finance, and loyalty systems.

3.4.2. Ethereum

Ethereum is a decentralized, open-source blockchain platform that enables the development of smart contracts and decentralized applications. It is widely used for both public and enterprise blockchain solutions.

3.4.3. Corda

Corda is a permissioned distributed ledger platform designed for business applications requiring high privacy. It enables secure transactions between relevant parties only, making it ideal for financial services, healthcare, and supply chain systems.

3.4.4. Quorum

Quorum is a permissioned blockchain platform derived from Ethereum, designed for enterprise use. It provides enhanced privacy, scalability, and faster consensus mechanisms, making it suitable for financial institutions and regulated environments.

3.5. Proposed System Architecture

The proposed system combines blockchain with a layered microservices web backend, where user management, blockchain interaction, and audit logging run as separate services. This makes the system easier to manage and scale, while ensuring that all actions are securely recorded in a decentralized and tamper-proof way using a consensus mechanism that validates data across all network nodes.

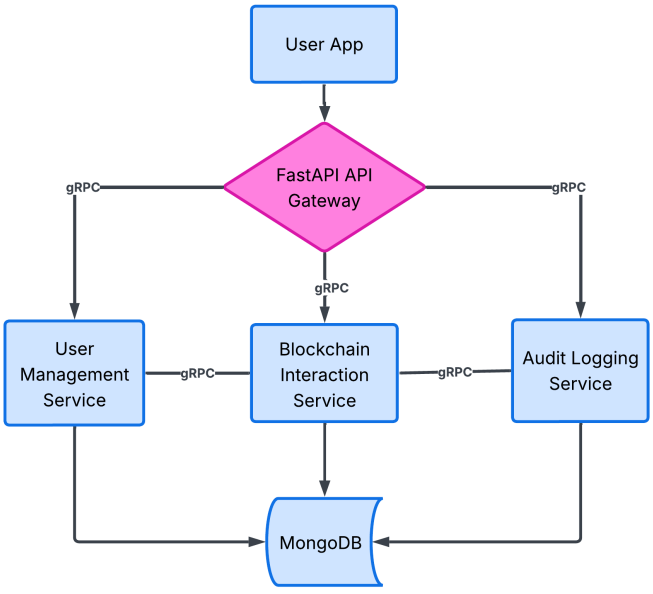

Figure 1. Blockchain Microservices Architecture.

Communication between services is handled using the gRPC framework, which enables fast and reliable communication between different services. The API layer is built with FastAPI to handle authentication, authorization,

and request routing. Data is stored in MongoDB, including user profiles, authentication records, verification results, and audit logs, which makes the system flexible and ensures high availability.

4. AI Verification Signal

4.1. Overview of the Dataset

For this research, experimental evaluation is carried out on handwritten and AI-generated signatures, using a method that can be applied to different types of documents and images. Handwritten signatures were created using real ink pens and do not belong to any specific individual, ensuring privacy. They were written on commonly used financial and official papers to better reflect real-world conditions. AI-generated signatures were created using generative AI models.

Table 1. *Handwritten and AI-Generated Signature Dataset.*

Signature Type	Count	Method / Tool
Handwritten	4550	Handwritten using ink pens on A4 paper and official forms.
AI-Generated	4550	OpenAI's GPT image models, Nano Banana 2, Qwen image generation models.

Variations in pen color, paper surface, and real-world document conditions can affect the performance of signature verification systems. To handle this, a preprocessing step and a simple detection model are used to filter out inputs that do not meet the expected conditions before classification.

Incorporating more diverse documents, handwriting styles, paper types, surface textures, and imaging conditions would improve robustness and generalization.

The dataset, created in this research, is released as an open-source resource under the MIT License at https://huggingface.co/datasets/Tapendra/Handwritten_AI_Signatures. It is split into 70% training, 15% validation, and 15% testing, and requires proper attribution for any use.

4.2. Input Size Standardization

To ensure consistent input, all signature images are resized to a fixed size while preserving aspect ratio, reducing scale variation and distortion.

Let W and H be the original width and height, and S the target model input size. The scaling factor is:

$$\text{scale} = \frac{S}{\max(W, H)}$$

The resized width W' and height H' are then:

$$W' = W \cdot \text{scale}, \quad H' = H \cdot \text{scale}$$

Padding is applied to the shorter side to form a square image, preserving shape and ensuring consistent input size. White padding is used to match the paper background and maintain natural appearance [24, 25].

4.3. Input Image Normalization

The input image I is converted into a tensor T by scaling pixel values from $[0, 255]$ to $[0, 1]$:

$$T = \frac{I}{255}$$

This scaled tensor is then channel-wise normalized to match the distribution of pretrained deep learning models.

$$T'_c = \frac{T_c - \mu_c}{\sigma_c}$$

where $c \in \{R, G, B\}$, and μ_c, σ_c are the channel-wise mean and standard deviation of the pretrained dataset [26].

4.4. ResNet50 Architecture

In this research, the ResNet-50 deep learning model was employed as a supporting signal for AI-based verification. In very deep networks, gradients become weak during backpropagation, leading to the vanishing gradient problem and limiting learning in early layers. ResNet-50 solves this using skip connections that improve gradient flow and enable residual learning for stable training [27–29].

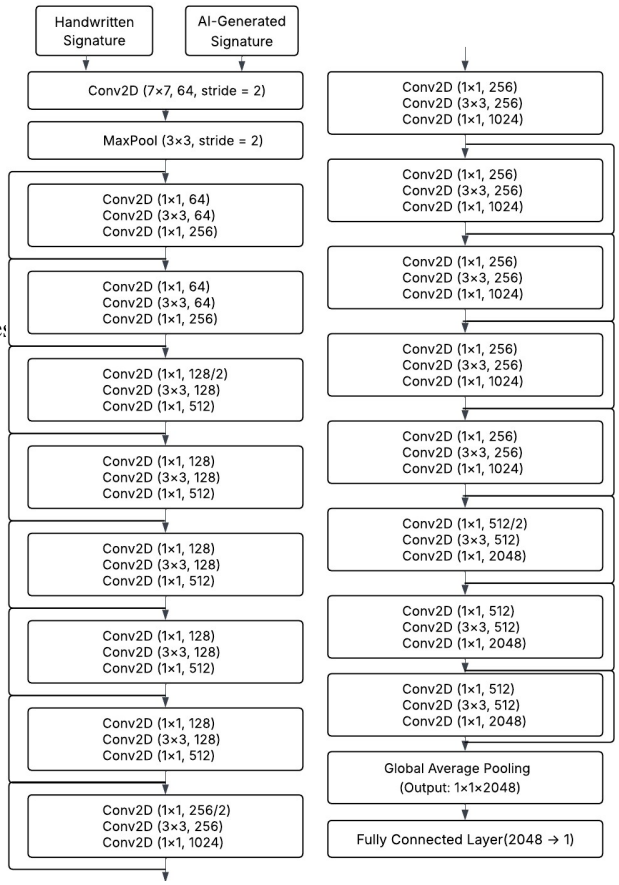

Figure 2. *ResNet architecture for signature verification.*

The output spatial dimensions are computed as:

$$O = \left\lfloor \frac{N + 2P - K}{S} \right\rfloor + 1$$

where N: input size, K: kernel size, P: padding, S: stride and O: output size

A kernel is a single 2D weight matrix applied to one input channel, while a filter is a collection of kernels across all input channels that produces one feature map.

An identity block is a component where the input and output dimensions remain the same. It uses skip connections to add the input directly to the output of convolution layers. It can be expressed as $y = F(x) + x$, where x is the input and $F(x)$ is the output of the convolution operations.

To reduce the feature map size, stride 2 is used, which changes the output dimensions. To handle this in the skip connection, a 1×1 convolution is applied in the shortcut path to match dimensions before addition. It is expressed as $y = F(x)+W_s x$, where $F(x)$ is the main path output and $W_s x$ is the shortcut connection output using a 1×1 convolution.

4.5. Training and Validation Loss

The model is trained using Binary Cross-Entropy with Logits Loss, which combines a sigmoid layer and BCELoss into a single numerically stable operation. This allows the raw output logits to be used to compute the loss with respect to the ground-truth labels, rather than applying a plain Sigmoid followed by a BCELoss. The log-sum-exp trick is applied to achieve a numerically stable form of the loss function [30].

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\sigma(\hat{y}_i)) + (1 - y_i) \log\big(1 - \sigma(\hat{y}_i)\big) \right]$$

where N is the number of samples, $\hat{y}_i \in \mathbb{R}$ is the raw logit output for sample i, $y_i \in \{0, 1\}$ is the corresponding ground-truth binary label, and $\sigma(\hat{y}_i) = \dfrac{1}{1 + e^{-\hat{y}_i}}$.

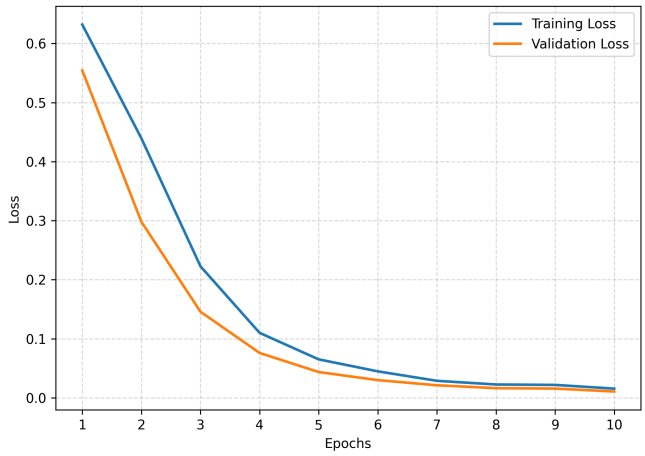

Figure 3. *Training and validation loss curve.*

The model is optimized using the AdamW optimizer with a learning rate $= 5 \times 10^{-6}$ and weight decay $= 1 \times 10^{-4}$.

The progressive reduction in both training and validation losses across epochs reflects stable learning dynamics and confirms effective model convergence without noticeable divergence.

4.6. Evaluation Metrics

In AI-generated and handwritten signature verification, TP represents correctly classified AI-generated signatures, TN represents correctly classified handwritten signatures, FP represents handwritten signatures misclassified as AI-generated, and FN represents AI-generated signatures misclassified as handwritten [31].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

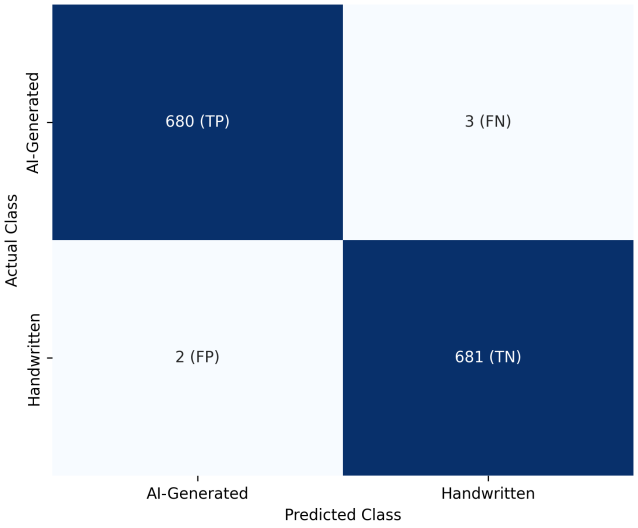

Figure 4. *Confusion Matrix for Performance Evaluation.*

The threshold of 0.45 was selected as it achieved the highest F_1-score within the tested range of threshold values. The model achieves an accuracy of 0.9963, precision of 0.9971, recall of 0.9956, and an F_1-score of 0.9963, showing consistent performance in classifying AI-generated and handwritten signatures.

4.7. ROC–AUC Curve

The Receiver Operating Characteristic (ROC) curve shows how well the model distinguishes between AI-generated and handwritten signatures at different classification thresholds.

The Area Under the Curve (AUC) summarizes this performance into a single value [32].

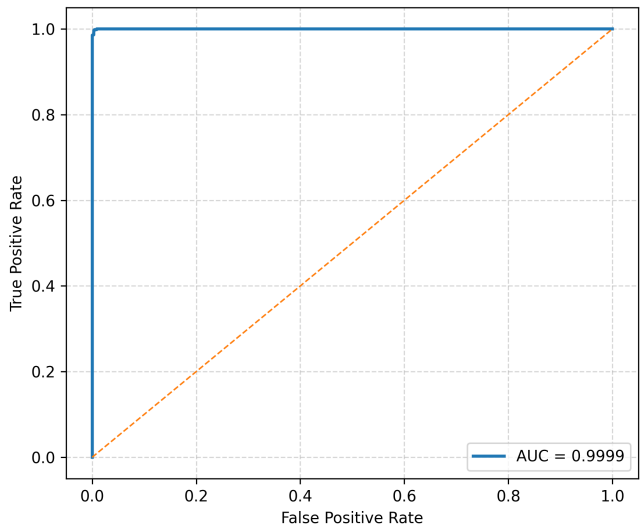

Figure 5. *ROC Curve and AUC Evaluation.*

The curve shows that the model performs consistently across different thresholds, and the high AUC value indicates good separation between the two classes.

5. Discussion and Future Work

Variations in pen color, paper surface, and real-world document conditions can affect the performance of signature verification systems. To handle this, preprocessing techniques and detection models are used to filter out inputs that do not meet the expected conditions before classification.

Recent advances in generative image models can produce highly realistic document images that closely resemble genuine ones. This makes it increasingly difficult to distinguish real document images from AI-generated forgeries as visual quality continues to improve. As a result, reliable detection becomes more challenging, highlighting the need for more robust and secure verification systems.

Future work will focus on extending the blockchain architecture across multiple distributed servers for real-world decentralized deployment. In addition, advanced models such as Vision Transformers (ViT) and hybrid CNN-Transformer architectures will be evaluated for verifying AI-generated documents and images using larger and more diverse datasets.

6. Results and Conclusions

The proposed blockchain microservices architecture with layered separation of user management, blockchain interaction, and audit logging services, utilizing gRPC and FastAPI, demonstrates how verification transactions can be recorded immutably with complete audit trails and tamper-proof evidence for digital identity verification. This ensures transparency, traceability, and trust in the verification process.

AI is used as a supporting signal for verification, where the ResNet-50 model shows strong performance in distinguishing AI-generated signatures from handwritten ones. The results show consistent performance across evaluation metrics such as precision, recall, F1-score, and ROC-AUC, confirming the effectiveness of the model. However, AI-based verification alone has inherent limitations, particularly in generalization to unseen generative models and evolving forgery techniques. These challenges highlight the need for complementary mechanisms to enhance reliability in real-world deployment scenarios, motivating a hybrid system that integrates blockchain technology with artificial intelligence for robust digital identity verification and signature forgery prevention.

ORCID

0009-0005-0355-5014 (Tapendra Baduwal)

Abbreviations

AI	Artificial Intelligence
CNN	Convolutional Neural Network
ViT	Vision Transformer
PRNU	Photo-Response Non-Uniformity
CFA	Color Filter Array
gRPC	Google Remote Procedure Call
ICOs	Initial Coin Offerings
PoW	Proof of Work
PoS	Proof of Stake
DIDs	Decentralized Identifiers
KYC	Know Your Customer
AML	Anti-Money Laundering
ROC	Receiver Operating Characteristic Curve
AUC	Area Under the ROC Curve

Author Contributions

Tapendra Baduwal: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Visualization, Validation, Writing - original draft, Writing - review & editing

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] J. Zuo, H. Deng, H. Zhou, J. Zhu, Y. Zhang, Y. Zhang, Y. Yan, K. Huang, W. Chen, Y. Deng, R. Jin, N. Sang, and C. Gao, "Is Nano Banana Pro a Low-Level Vision All-Rounder? A Comprehensive Evaluation on 14 Tasks and 40 Datasets," https://doi.org/10.48550/arXiv.2512.15110

[2] H. Yang, Y. Yang, R. Zhang, and L. Pan, "A Preliminary Study for GPT-4o on Image Restoration," https://doi.org/10.48550/arXiv.2505.05621

[3] S. Susnjara and I. Smalley, "Blockchain," *IBM Think*, 2026. https://www.ibm.com/think/topics/blockchain

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. https://bitcoin.org/bitcoin.pdf

[5] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," White Paper, 2014.

[6] C. Fisch, "Initial coin offerings (ICOs) to finance new ventures," *Journal of Business Venturing*, vol. 34, pp. 1–22, Jan. 2019, https://doi.org/10.1016/j.jbusvent.2018.09.007

[7] IBM, "What is Blockchain and AI?," *IBM Think*, 2026. https://www.ibm.com/think/topics/blockchain-ai

[8] A. Nehe, A. Bhagwat, V. Lamkhade, S. Kshirsagar, R. Kadlag, and A. Shinde, "Blockchain Based Identity Verification System," *World Journal of Pharmaceutical Science and Research*, vol. 4, pp. 981-997, Nov. 2025, https://doi.org/10.5281/zenodo.17617605

[9] C. J. Zhang, A. Q. Gill, B. Liu, and M. J. Anwar, "AI-based Identity Fraud Detection: A Systematic Review," Jan. 2025. https://doi.org/10.48550/arXiv.2501.09239

[10] Babaei, R., Cheng, S., Duan, R., and Zhao, S. (2025). Generative Artificial Intelligence and the Evolving Challenge of Deepfake Detection: A Systematic Analysis. Journal of Sensor and Actuator Networks, 14(1), 17. https://doi.org/10.3390/jsan14010017

[11] Rahman, M. M., Tonmoy, M. T. K., Shihab, S. R., and Farhana, R. (2023). Blockchain-based certificate authentication system with enabling correction. https://doi.org/10.48550/arXiv.2302.03877

[12] Lopes, A. D., Mello, T., and Bezerra, W. R. (2025). Digital identity management system with blockchain: An implementation with Ethereum and Ganache. https://doi.org/10.48550/arXiv.2507.21398

[13] Vinogradov, A. (2025). Can generative models actually forge realistic identity documents? https://doi.org/10.48550/arXiv.2601.00829

[14] Y. Hao and Z. Zheng, "Research on detecting AI-generated forged handwritten signatures via data-efficient image transformers", https://doi.org/10.1109/ACCESS.2025.3525808

[15] A. A. Abdirahma, A. O. Hashi, M. A. Elmi, and O. E. R. Rodriguez, "Advancing handwritten signature verification through deep learning: A comprehensive study and high-precision approach. https://doi.org/10.14445/22315381/IJETT-V72I4P109

[16] N. Gupta, "Security risks of generative AI in financial systems: A comprehensive review," *World Journal of Information Systems*, vol. 1, no. 3, pp. 17–24, 2025. https://doi.org/10.17013/wjis.v1i3.16

[17] H. Natarajan, S. Krause, and H. Gradstein, "Distributed ledger technology (DLT) and blockchain," World Bank FinTech Note No. 1, International Bank for Reconstruction and Development / The World Bank, Washington, DC, USA, 2017.

[18] C. Nguyen, H. T. Thai, D. Nguyen, D. Niyato, H. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," *IEEE Access*, pp. 1–1, 2019, https://doi.org/10.1109/ACCESS.2019.2925010

[19] S. N. Khan et al., "Blockchain smart contracts: Applications, challenges, and future trends, *Peer-to-Peer Networking and Applications*," vol. 14, no. 5, pp. 2901–2925, 2021. https://doi.org/10.1007/s12083-021-01127-0

[20] W. Zhou, D. Lyu, and X. Li, "Blockchain security based on cryptography: A review," 2025. https://doi.org/10.48550/arXiv.2508.01280

[21] Susnjara, S., and Smalley, I., "What is Blockchain?," *IBM Think*, 2026. https://www.ibm.com/think/topics/blockchain

[22] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," 2017. https://doi.org/10.1109/BigDataCongress.2017.85

[23] Amazon Web Services, "What is blockchain technology?," https://aws.amazon.com/what-is/blockchain/

[24] Shubham Randive, "How to Handle Variable-Size Images in Deep Learning Batches - A Practical Guide to Batching Strategies for Training Vision Models," https://medium.com/@randiveshubham3/how-to-handle-variable-size-images-in-deep-learning-batches-19fd6880a4e3

[25] Timothy M, "What is Image Resizing? A Computer Vision Guide," https://blog.roboflow.com/image-resizing

[26] PyTorch,"Transforming images, videos, boxes and more." https://docs.pytorch.org/vision/stable/transforms.html

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," 2015. https://doi.org/10.48550/arXiv.1512.03385

[28] Aditi Rastogi,"ResNet50," Dev Genius Blog, https://blog.devgenius.io/resnet50-6b42934db431

[29] Abirami Vina, "What is ResNet-50 and what is its relevance in computer vision?" Ultralytics Blog. https://www.ultralytics.com/blog/what-is-resnet-50-and-what-is-its-relevance-in-computer-vision

[30] PyTorch Developers. BCEWithLogitsLoss. PyTorch Documentation, version 2.10.0 (21 March 2026).

[31] Zeljko Vujovic, Classification Model Evaluation Metrics, International Journal of Advanced Computer Science and Applications, vol. 12, pp. 599–606, 2021. https://doi.org/10.14569/IJACSA.2021.0120670

[32] Aniruddha Bhandari, Guide to AUC ROC Curve in Machine Learning, Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning