Application of ZKML for Unpredictive Epidemic Response

Jin Ah Seo[1], Kun Hwa Lee[2], Vijayan Sugumaran[3], Jo Yeon Park[1] and Soo Yong Park[1,*]

[1]*Sogang University, Department of Computer Science, Seoul, South Korea*
[2]*Seoul National University, Department of Electrical and Computer Engineering, Seoul, South Korea*
[3]*Institute for Data Science, School of Business Administration, Oakland University, Rochester, Michigan, USA*
E-mail: jinah12@sogang.ac.kr; lkh0107@snu.ac.kr; sugumara@oakland.edu; hpjoanne@sogang.ac.kr; sypark@sogang.ac.kr
[*]*Corresponding Author*

Received 17 May 2025; Accepted 17 March 2026

Abstract

We build and evaluate a concrete Zero-Knowledge Machine Learning (ZKML)-based pipeline for epidemic diagnosis and show that it can enforce computational integrity without exposing raw medical data in a Web3 setting. In response to security challenges posed by centralized data handling in medical AI applications, particularly during public health crises such as COVID-19, ZKML offers a privacy-preserving alternative by combining machine learning and Zero-Knowledge Proofs (ZKP). We experimentally applied ZKML to a CNN (Convolutional Neural Networks)-based COVID-19 diagnostic model, achieving 87% accuracy and 0.35 loss. All proof generation and verification processes were executed entirely off-chain, with the verified outputs represented as committed public_vals recorded on-chain via smart contracts. To ensure authenticity, the system enforces dual ECDSA signature verification from both the model provider and the data provider. This

Journal of Web Engineering, Vol. 25_5, 889–914.
doi: 10.13052/jwe1540-9589.2556

mechanism prevents unauthorized submissions and confirms the validity of the result before it is stored on-chain. The system was tested under both normal and adversarial conditions, demonstrating robust and reliable operation. By enabling decentralized trust and self-sovereign control over data, this architecture aligns well with Web3 principles. The results indicate that ZKML can support the development of privacy-preserving and verifiable AI systems.

Keywords: Zero-knowledge proof, machine learning, zero-knowledge machine learning, privacy, medical.

1 Introduction

This study proposes a Zero-Knowledge Machine Learning (ZKML)-based machine learning procedure that ensures computational integrity while protecting the privacy of medical data in a Web3 environment. During COVID-19, hospitals relied heavily on automated triage and imaging tools, often built on machine learning models. These systems had to operate under tight time and resource constraints. Consequently, ML (Machine Learning) models have played a crucial role in various fields, including disease diagnosis, infection spread prediction, and medical resource allocation [1–3], in particular, deep learning-based diagnostic models using chest X-ray (CXR) and CT images in pandemic response, assisting medical professionals in making faster and more accurate decisions [4]. However, as ML models are applied in medical data environments, concerns about privacy protection and reliability have become increasingly critical.

Since ML models require large-scale medical datasets for training and prediction, several security and privacy risks arise in healthcare environments. In particular, medical datasets contain highly sensitive information, including electronic health records, medical images, and clinical reports. Unauthorized access or data breaches involving these data may lead to serious privacy violations and misuse of personal medical information [5]. In addition, machine learning models used in healthcare may suffer from algorithmic bias originating from the training data. If the datasets used to train the model contain imbalanced or inaccurate records, particularly those related to sociodemographic characteristics such as race, gender, age, or socioeconomic status, the resulting models may perpetuate existing healthcare

disparities. Such biases can lead to unequal diagnostic outcomes or treatment recommendations across different patient populations [6]. Finally, another challenge in applying machine learning to healthcare arises from the limited interpretability of many predictive models. While such models can achieve strong diagnostic performance, their internal reasoning mechanisms are often difficult to interpret, which may limit the level of trust clinicians place in automated decision-support systems [7].

Recently, the ZKML framework has been proposed as a solution to these challenges [8]. ZKML integrates Zero-Knowledge Proofs (ZKP) into ML workflows, enabling the verification of computational integrity while preserving the confidentiality of sensitive medical data. In particular, a Web3-based medical data environment requires a shift from centralized server-based data management to decentralized systems for enhanced security, and ZKML presents an effective alternative in such environments.

This study aims to apply the existing ZKML framework to a Web3-based medical data environment and evaluate its practicality in a COVID-19 diagnostic system. Specifically, we investigate whether ZKML can protect medical data privacy while maintaining the predictive performance of CNN (Convolutional Neural Networks)-based diagnostic models and assess whether proof generation and verification processes are practical for real-world medical AI applications. Furthermore, this study explores the integration of ZKML within Web3-based medical data-sharing systems to propose a scalable verification framework for medical AI. Unlike prior studies that mainly focus on theoretical constructions of ZKML, this work provides an end-to-end experimental validation of a ZKML-enabled medical AI pipeline integrated with Web3 infrastructure.

1.1 Research Contributions

This study presents a novel approach for leveraging ZKML-based COVID-19 diagnostic systems to ensure both medical data privacy and computational integrity. While prior research has explored various techniques to enhance privacy and reliability in medical AI systems, there has been a lack of empirical studies evaluating the practical applicability of machine learning models integrated with ZKP. This study demonstrates that ZKML can be effectively applied in medical AI environments, maintaining data security while preserving predictive accuracy. The primary contributions would be as follows.

First, this study empirically evaluates the feasibility of integrating ZKML into a CNN-based medical diagnostic workflow. By integrating a ZKP-based verification system into a CNN-based COVID-19 diagnostic model, this research examines a framework that enables secure medical diagnosis without directly exposing sensitive patient data. The findings confirm that ZKML can simultaneously ensure data privacy and model reliability in AI-driven diagnostics.

Second, this study analyzes the proof generation and verification procedures in a ZKML-based inference pipeline. By utilizing an existing ZKML framework, the research assesses the computational cost associated with proof generation and examines whether cryptographic proofs can be independently verified by external entities. Unlike previous studies that primarily focus on theoretical aspects, this study offers a practical evaluation using real-world medical data.

Third, the study evaluates the effect of ZKML integration on CNN model performance and inference reliability. Using a publicly available medical dataset containing COVID-19, Lung Opacity, Normal, and Viral Pneumonia [24] cases, the study demonstrates that the CNN-based model maintains an accuracy of 87% within the ZKML environment while achieving a loss value of 0.35, confirming stable performance. These results suggest that the integration of ZK proofs does not significantly degrade model accuracy while enhancing computational verifiability.

In particular, this study performed the generation and verification of ZKP entirely off-chain. The public_vals was stored on-chain in the form of a hash value, and the authenticity was verified by checking the signature of the on-chain hash through a smart contract. By verifying the signature, the integrity and trustworthiness of the user's input were ensured. The system was tested with both legitimate users and inputs intentionally manipulated by malicious users. Furthermore, an analysis of computational overhead confirmed the practical applicability of ZKML in Web3-based medical environments. The study also demonstrated that proofs could be independently verified by external verifiers, thereby supporting the transparency and trustworthiness of AI systems.

Finally, this study introduces a Web3-based verification architecture that combines off-chain ZK proof generation with on-chain signature validation. This architecture enables an auditable and privacy-preserving medical AI pipeline and provides a scalable verification framework that ensures computational integrity in Web3 medical environments, thereby supporting the development of trustworthy medical AI systems.

2 Background and Related Work

2.1 ZKP & ZKML

ZKPs enable a prover to convince a verifier that a computation has been executed correctly while keeping the underlying input data confidential [9]. ZKP must satisfy three key properties: completeness, soundness, and zero-knowledge. The most widely used implementations of ZKP include zk-SNARK and zk-STARKs [10]. ZKP is defined formally as a protocol between the prover p and the verifier v. p wants to prove the possession of witness w for some input ϕ, where $(\phi, w) \in R$ is the pair of the public statement (input and/or output) and secret witness, and R is the relation. p generates a proof π and shares it with v, which checks its validity without learning anything about w. w contains all the relevant values of the computation, initial input values, intermediate values of the computation itself and output values [11].

Traditional ML models typically train and make predictions on centralized servers, which may expose sensitive medical information. Additionally, a major challenge is the lack of independent verification methods for ensuring the integrity of prediction results generated on centralized servers. To address these issues, ZKML has emerged.

ZKML is a technique to convert machine learning models into mathematical representations suitable for zero-knowledge proofs. Specifically, ML computations are typically represented using constraint systems such as Rank-1 Constraint Systems (R1CS) and polynomial representations, as discussed in the ZKP section. These transformed computations are then used to generate proofs using zk-SNARKs or zk-STARKs, ensuring that the AI model has executed correctly. Once the proof is generated, external validators can independently verify the accuracy of the computation, which helps ensure the security and reliability of the AI Model. Recent studies have proposed applying ZKML to provide proofs along with model predictions, allowing external verifiers to independently confirm the reliability of the results. Chen et al. has implemented TFLite models into fixed-point gadgets so that CNNs and transformers can be proven in zero knowledge [8]. Li et al. has applied to a lightweight zero-knowledge aggregation protocol for FL [12]. Chen et al. has compared trusted execution environments, secure multiparty computation, differential privacy, and fully homomorphic encryption and ZKP [13].

To overcome the privacy limitations of centralized machine learning systems, several privacy-preserving learning paradigms have been investigated

Table 1 Comparison of privacy-preserving machine learning approaches including ZKML

Method	Privacy	Verifiability	Computation Cost
FL (Federated Learning)	Medium	Low	Medium
DP (Differential Privacy)	High	Low	Low
MPC (Multi-Party Computation)	High	Medium	High
ZKML	High	High	High

in the literature. Representative approaches include Federated Learning (FL), Differential Privacy (DP), and Secure Multi-Party Computation (MPC).

As shown in Table 1, privacy-preserving machine learning approaches including FL, DP, and MPC exhibit different trade-offs between privacy protection, verifiability, and computational cost. FL and DP mainly focus on protecting sensitive data but do not provide cryptographic verification of inference results. MPC enables secure collaborative computation but typically introduces significant computational overhead. In contrast, ZKML enables verifiable AI inference by generating ZKP for machine learning computations.

2.2 zk-SNARKs

ZKP typically implemented zk-SNARKs and zk-STARKs in two ways. zk-SNARKs are widely used due to their compact proof size and efficient verification, although they typically rely on a trusted setup phase. It is used in a privacy-oriented blockchain such as Zcash. zk-STARKs do not require a trusted setup and are suitable for large-scale data processing. It is mainly used in blockchain scalability solutions [14].

Formally, a proof for a relation R is a protocol where ρ convinces ν that there exists a witness ω such that $\mathrm{R}(\chi, \omega) = 1$, where χ is called the instance and w is a witness for χ. In the case of non-interactive proofs, three polynomial-time algorithms define the proof system:

Setup$(1\lambda, R) \rightarrow (\rho, \nu\kappa)$: Generates the public parameters (crs) and a $\nu\kappa$ (verification key) for relation R given a security parameter λ.

Prove$(\rho, \chi, \omega) \rightarrow \pi$: Produces a proof π if R $R(\chi, \omega) = 1$.

Verify$(\nu\kappa, \chi, \pi) \rightarrow \{0, 1\}$: Verifies whether π is a valid proof for instance χ [15].

zk-SNARKs generate and verify proofs using polynomial verification and elliptic curve operations. The depiction in Figure 1 visualizes the entire

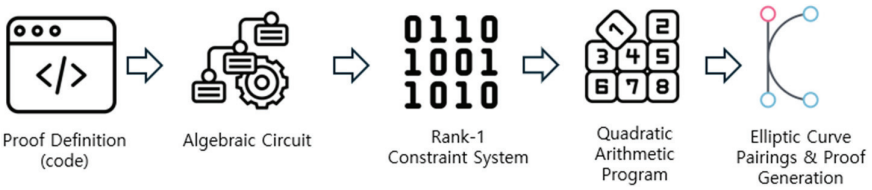

Figure 1 zk-SNARKS flow.

computational process of zk-SNARKs, illustrating the key steps involved in the proof generation and verification.

First, the computation to be proved is mathematically expressed in polynomial form and then converted into an algebraic circuit composed of logical gates. Next, the circuit is transformed into a matrix representation, defining constraint conditions using three matrices to formulate the equation. These defined circuit constraints are then converted into a Quadratic Arithmetic Program (QAP), which allows for the mathematical verification of computational integrity.

In the core computational process, polynomial evaluations are performed at specific points, and in the final stage, elliptic curve pairing operations are used to generate the zero-knowledge proof. In zk-SNARKs, an invariance verification process ensures that the proof satisfies the given constraints before generating a proof value that allows an independent verifier to confirm its validity.

This study aims to leverage zk-SNARKs polynomial verification and elliptic curve operations to build a reliable framework for medical AI environments.

2.3 Machine Learning

Machine learning refers to a computational approach that enables systems to build predictive models automatically from data. Recent advances in machine learning have explored lightweight architectures, robustness, and weakly supervised learning approaches [19–23]. It allows computers to automatically learn and improve from experience without being explicitly programmed [16]. The COVID-19 pandemic has made the importance of AI technologies, particularly deep learning-based diagnostic systems, which have been actively utilized in medical image analysis [1, 2]. Diagnostic models leveraging CNN have proven effective in detecting COVID-19 infections based on CXR and CT images, offering faster and more accurate

diagnoses compared to traditional diagnostic methods. However, when these ML models are applied in medical environments, the storage and processing of patients' sensitive data on centralized servers raise critical concerns regarding privacy protection and data integrity [3].

Apostolopoulos and Mpesiana investigated the effectiveness of deep learning architectures for detecting COVID-19 using CXR images by evaluating the performance of state-of-the-art convolutional neural network models trained on datasets containing pneumonia, COVID-19, and normal cases [1]. Tartaglione et al. examined the potential of limited COVID-19 X-ray datasets and CXR imaging for early screening of COVID-19 patients and demonstrate that deep learning models can be evaluated using sensitivity and specificity metrics [2]. Chowdhury et al. conducted a study to investigate the usefulness of artificial intelligence (AI) and proposed an automated detection approach for identifying COVID-19 pneumonia from CXR images. The study showed that applying pre-trained deep learning algorithms can improve detection accuracy [3].

Recent studies have explored blockchain-based frameworks to enhance the security and management of healthcare data in distributed systems. Alkhalil et al. proposed a blockchain-enabled mobile healthcare (mHealth) framework that integrates Ethereum smart contracts and IPFS storage to securely manage medical data in decentralized environments [17]. Song et al. introduced a peer-to-peer federated learning framework that supports collaborative model training across distributed edge environments while preserving local data privacy and improving communication efficiency in heterogeneous systems [18]. However, although these decentralized approaches improve data security and privacy protection, they do not guarantee the verifiability of machine learning computations, which highlights the need for secure frameworks such as ZKML.

3 Design and Implementation of the ZKML Framework for Epidemic Response

3.1 ZKML-Based Machine Learning Procedure

Figure 2 illustrates the overall workflow of the proposed ZKML-based medical diagnostic framework, including data submission, model inference, proof generation, verification, and on-chain registration. This framework adopts a fully off-chain zk-SNARK architecture, where both proof generation and verification are performed entirely off-chain. The only data submitted

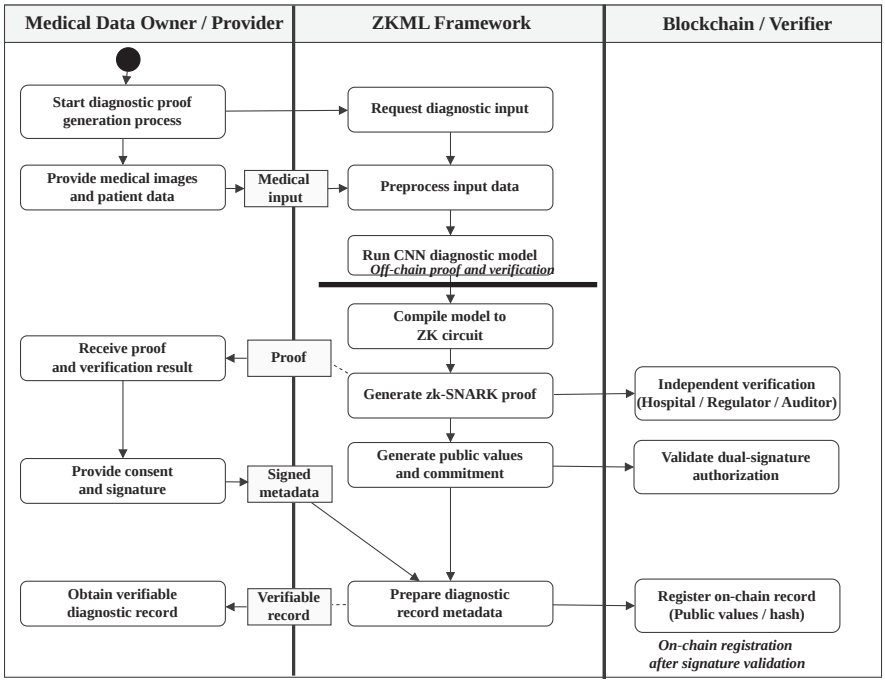

Figure 2 COVID-19 diagnostic framework based on ZKML.

on-chain is a public commitment derived from the inference computation, enabling verifiability without revealing sensitive information. To protect patient privacy, raw medical data such as CXR or CT scan images is never uploaded or stored externally. Instead, it is processed locally within a trusted environment and used as input to a CNN based AI model. The model, trained to classify patients into four categories COVID-19, Viral Pneumonia, Lung Opacity, and Normal is compiled into a ZK circuit using a ZKML compiler. This compilation enables the inference to be performed as part of a ZKP, ensuring that predictions can be mathematically verified without exposing the model parameters or input data. During inference, the ZK circuit processes the input, generates a prediction, and constructs a zk-SNARK proof that attests to the correctness of the computation. This proof is then verified off-chain by a trusted verifier module. Upon successful verification, the system extracts the public_vals. These public_vals are then submitted to the blockchain via a smart contract as a public record of the verified computation.

To ensure the authenticity and integrity of these on-chain records, the smart contract requires two cryptographic signatures. The model provider (e.g., hospital or AI solution vendor) signs the public_vals, confirming that the result was generated using an approved and registered model. The data provider (e.g., the patient or the data-owning hospital) also signs the same values, granting explicit consent to commit the result on-chain. These two ECDSA signatures are verified within the smart contract before any public value is stored. If both the model provider and data provider have signed the same public_vals, then the result is recorded on-chain. This dual-signature mechanism ensures that the data has not been tampered with and that both parties agree to the result being registered in the public domain.

When public appointments come on-chain, third parties, such as hospitals, regulators, or researchers, can recalculate and input inference using the same model to verify the accuracy of the results by ensuring that they match the on-chain public_vals. Because no raw data or proof is ever published, the privacy of sensitive patient information remains fully preserved.

Furthermore, the framework includes an on-chain model registration feature, allowing model publishers to record their deployment and update history. This provides additional transparency and helps ensure that only trusted AI models are used in clinical workflows.

In summary, the proposed framework combines off-chain zk-SNARK computation, on-chain cryptographic verification, and dual-party consent to deliver a secure, privacy-preserving, and trustworthy pipeline for AI-assisted medical diagnostics.

3.2 ZK Proof Generation and Verification

To guarantee the correctness of the CNN model's predictions, zk-SNARK proof is generated during the inference phase. This proof, denoted as P, formally expresses that the model executed its computations without modification or external interference. The verification process follows the equation.

$$\mathrm{P} = \mathrm{ZKP}(\mathrm{M}(\mathrm{X}) = \mathrm{Y})$$

where P represents the zk-SNARK proof, $\mathrm{M}(\mathrm{X})$ is the CNN model's inference process, and Υ is the classification result. The generated proof is then verified by an independent entity, such as a hospital, research institution, or regulatory authority, without requiring direct access to either the model's internal structure or the patient's medical data. The verification function is

as follows.

$$\mathrm{V}(\mathrm{P}, \mathrm{X}, \mathrm{Y}) \rightarrow \{0, 1\}$$

If the proof is valid, the verifier confirms that the model's prediction is trustworthy and has not been tampered with. The verification process can occur off-chain for efficiency or be recorded on-chain via smart contracts to provide an immutable audit trail.

4 Experimental Setup and Results

4.1 Experimental Setup

The experimental setup was designed to systematically evaluate both model performance and proof verification speed under realistic conditions. The CNN model was trained on a large dataset of CXR images, and its predictive accuracy was assessed across four distinct diagnostic categories. The trained model was then converted into a zero-knowledge circuit, allowing inference computations to be transformed into zk-SNARK proofs. The proof verification process was subsequently tested to measure its computational efficiency and determine its feasibility for real-time medical applications.

The evaluation process involved two primary components

1. Model Performance Analysis: Assessing the classification accuracy, precision, recall, and F1-score of the CNN-based COVID-19 diagnosis model.
2. ZK Proof Computation Analysis: Measuring the time required for proof generation and verification, evaluating its impact on inference efficiency.

By examining these two aspects, the study sought to determine whether ZKML could be practically deployed in medical settings where both privacy and verifiability are critical concerns. To ensure reproducibility, the complete implementation details, including dataset preprocessing, CNN model training scripts, and ZK proof generation algorithms, have been made publicly available at https://github.com/BaSELab-ZKML/codes. This repository provides all necessary configurations for researchers interested in replicating or extending the study.

The experiments were conducted on a workstation with an AMD Ryzen Threadripper PRO 5975WX CPU (32 cores, 64 threads), 251 GiB of RAM, and 31 GiB of swap memory. The system was running Ubuntu 22.04.5 LTS with kernel version 5.15.0-136-generic. The software environment included

Python 3.10.12, TensorFlow 2.21.0, Rust 1.94.0, and Cargo 1.94.0. The ZKML framework used in this study was zkml (commit 4378958).

4.2 Dataset and Model Training

To train a CNN-based epidemic diagnostic model, we used the publicly available CXR dataset [20]. The dataset includes four diagnostic categories: COVID-19, viral pneumonia, lung opacity, and normal cases, and consists of 42,330 X-ray images. To ensure consistency in the input size, all images were resized to 256 $\times$ 256 pixels. The images were converted from RGB to grayscale and pixel values were normalized to improve training stability. The dataset was evaluated by dividing it into 80% training data (33,864 images) and 20% test data (8,466 images). The CNN architecture consisted of three convolution layers (16, 32, 64 filters) and an output layer with SoftMax activation for four-class classification. To avoid overfitting, we introduced a dropout layer (ratio: 0.3) and optimized the model with a categorical cross-entropy loss function (learning rate: 0.001). After model training, the trained CNN model was integrated into the ZKML framework and converted into a zero-knowledge circuit to generate verifiable zk-SNARK proofs for prediction. Within the inference pipeline, we incorporated ZK proof generation to enable cryptographic validation of all diagnostic decisions without having to disclose sensitive patient data. The model was trained for multiple epochs using the Adam optimizer with a batch size of 32, and early stopping was applied to prevent overfitting.

4.3 Performance Evaluation of ZK Proof Computation

To assess the feasibility of deploying ZKML-based medical AI models in real-world clinical environments, an in-depth evaluation of the model performance and proof verification speed was conducted.

4.3.1 CNN model performance evaluation

The classification accuracy of the CNN model was measured using standard performance metrics such as accuracy, precision, recall, and F1-score. The results demonstrated that, for comparison, the baseline CNN model without ZKML integration achieved an accuracy of 88.2%, while the CNN model integrated with ZKML maintained an accuracy of approximately 87–88%. This result confirms that the ZK proof generation process does not significantly affect model prediction accuracy. Confirming its reliability for

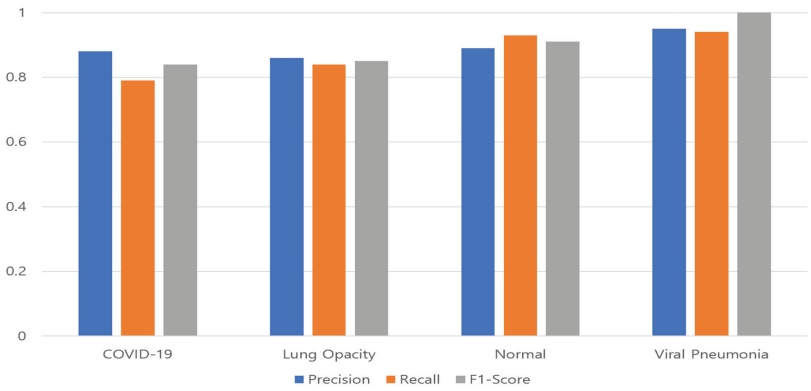

Figure 3 Performance evaluation metrics of the CNN model.

```
ZkmlVerifierRegistry
User1 Address: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
Fake Public Vals: this is a test public_vals
Computed Hash (Bytes32): 0xe7e486f13060bca24697f785930e62f83ec391597cb63cd6cd6194d2cd2057a4
Stored Hash from Contract: 0xe7e486f13060bca24697f785930e62f83ec391597cb63cd6cd6194d2cd2057a4
    ✓ should store public hash for the sender

=== [Valid Proof Test] ===
User1 Address: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
Original Data: zkml is cool
Computed Hash: 0xe97da9b49cf584440e6c2335fa81741bb493f3eac6dc3a7a9078dcbfb849701b
Verification Result (Correct Hash): true

Wrong Data Hash (for failure test): 0x4d51758d441de0e98e0a9dc17ff37bbaa98840044bd2515f85ee429035ba7f6c
Verification Result (Wrong Hash): false
    ✓ should verify correct hash


2 passing (530ms)
```

Figure 4 On-chain hash verification result for valid and invalid ZKML inputs.

COVID-19 diagnosis. As shown in Figure 3, the F1-score for the Viral Pneumonia class was 0.95, the highest among the four diagnostic categories, whereas the COVID-19 class recorded an F1-score of 0.84. The Normal class exhibited a precision of 0.89 and recall of 0.93, indicating that the model effectively distinguished between healthy and infected cases. These results indicate that integrating ZK proofs maintains the predictive performance of the CNN model while enabling cryptographic verification of inference results.

4.3.2 On-chain-based signature verification in ZKML

In this study, the system architecture has been implemented in which the generation and verification of ZKP are performed entirely off-chain, while the hash value of the public_vals is recorded on-chain. The authenticity of this data is verified by checking the user's digital signature on the on-chain

```
ZKMLPublicValsStorage
  ✓ should correctly store the list of approvers
  ✓ should not store public_vals without any approval
  ✓ should store public_vals after 2 valid approvals
  ✓ should reject duplicate approvals from the same approver
  ✓ should reject approval from non-approver
  ✓ should reject additional approvals after public_vals is stored


6 passing (1s)
```

Figure 5 Testing signature verification within a smart contract.

hash via a smart contract. This architecture was validated through practical testing using both legitimate and maliciously manipulated inputs. The results confirmed that the system accurately distinguished between valid and invalid inputs. Figure 4 illustrates the validation process of on-chain hash values generated from ZKML inference outputs. The experiment demonstrates that the smart contract correctly distinguishes between valid and manipulated inputs.

As shown in Figure 5, once the smart contract is deployed, an Approver List is initialized. For a specific public_vals to be stored on-chain, valid digital signatures from at least two registered approvers are required. The system is designed to automatically reject submissions that include no approval, duplicate signatures from the same approver, signatures from non-approved entities, or any excessive approvals. Figure 5 presents the signature verification mechanism implemented within the smart contract. Only transactions containing valid signatures from authorized approvers are accepted for on-chain registration. This enforcement mechanism ensures the integrity and trustworthiness of the signature verification process.

Although the proof generation phase incurs some computational overhead, the verification stage remains highly efficient, relying only on lightweight cryptographic operations. As a result, diagnostic outputs can be verified rapidly, with minimal latency. These findings strongly support the feasibility of adopting ZKML as a privacy-preserving and verifiable AI diagnostic framework suitable for real-world medical applications.

4.3.3 Ensuring consistency of AI predictions

One of the key requirements for integrating ZKML into AI inference is ensuring that the predicted outputs remain unchanged after applying the proof generation process. To validate this, we compared the classification results of a standard CNN model with those of a CNN model integrated

Table 2 Output consistency between standard CNN and ZKML-integrated CNN

Label	Standard CNN Output	CNN+ZKML Output	Result
Covid-19	[0]	[0](0.97)	same
Lung Opacity	[1]	[1]	same
Normal	[2]	[2]	same
Viral Pneumonia	[3]	[3]	same

Table 3 Prediction consistency between CNN and CNN+ZKML (10 test samples)

Category	Ground Truth (GT)	CNN Prediction	CNN + ZKML
Samples	[COVID, COVID, COVID, Normal, Normal, Normal, Lung_Opacity, Lung_Opacity, Viral_Pneumonia, Viral_Pneumonia]	[COVID, COVID, COVID, Normal, Normal, Viral_Pneumonia, Normal, Lung_Opacity, Lung_Opacity, Viral_Pneumonia]	[COVID, COVID, COVID, Normal, Normal, Viral_Pneumonia, Normal, Lung_Opacity, Lung_Opacity, Viral_Pneumonia]
Consistency	–	–	All predictions identical between CNN and CNN+ZKML

with ZKML proof generation. Our goal was to determine whether the use of zk-SNARKs affects model predictions. To verify consistency, we conducted inference on the same dataset using a CNN model trained for COVID-19 diagnosis and ZKML. The classification labels were recorded and compared across both models. The results confirm that the integration of ZK proof generation does not impact the output consistency of the AI model. The model classifies CXR images into four diagnostic categories. The predicted labels are represented numerically as [0] COVID-19, [1] Lung Opacity, [2] Normal, [3] Viral Pneumonia.

This demonstrates that the ZK proof generation process does not introduce numerical variations or distortions in AI predictions. The results presented in Table 2 validate that ZKML maintains the integrity of AI model outputs while providing cryptographic verifiability. This ensures that AI-driven medical diagnostics can benefit from enhanced privacy and security without compromising prediction accuracy.

As shown in Table 3, the predictions generated by the CNN model and the CNN integrated with ZKML are identical for all 10 samples. This confirms that the ZK proof generation process does not alter the inference results of the CNN model.

Table 4 ZKML computation time analysis

Component	Average Time	Measurement Criteria
CNN inference	54.39 ms/image	CPU environment, average of 10 test samples
ZK proof generation	1587.86 s/sample (≈26.46 min)	Average proving time from time_circuit logs (10 runs)
ZK proof verification	10.62 ms/sample	Average verifying time from time_circuit logs (10 runs)

4.3.4 ZK proof computation time analysis

To evaluate the computational overhead introduced by the ZKML pipeline, we measured the CNN inference time, ZK proof generation time, and proof verification time under the same experimental environment. All measurements were repeated 10 times and the average values were reported to ensure measurement stability. The CNN inference time was measured in a CPU-based environment using 10 test samples. The proof generation and verification times were obtained from the time_circuit execution logs of the ZKML framework. The measured results are summarized in Table 4.

As shown in Table 4, the CNN inference time is relatively small, while the ZK proof generation process requires significantly more computation time. In contrast, proof verification remains extremely efficient, requiring only 10.62 ms on average. This result indicates that although proof generation introduces a considerable computational overhead, the verification stage remains lightweight and suitable for practical deployment scenarios where verification must be performed frequently.

5 Discussion

5.1 Limitations of This Study

This study aimed to ensure both privacy protection and verifiability of AI diagnostic results in a Web3-based medical environment by applying the ZKML framework. However, several limitations must be considered for its practical application in real-world Web3 systems.

First, ZK proof generation requires high computational resources, which can be a burden for real time diagnosis. During the proof generation process, it depends heavily on GPU performance, such as NVIDIA RTX 3090. Although the verification latency is relatively small, proof generation remains computationally intensive. In large-scale medical environments, parallel proof generation and circuit optimization techniques may be required to

support high-throughput clinical workflows. Therefore, the practical deployment of the proposed ZKML framework may be challenging in environments without sufficient computational resources.

Second, this system adopts a structure in which zero-knowledge proof generation and verification are performed off-chain, and the resulting hash value is recorded on-chain, with signature verification conducted via a smart contract. While this approach reduces the computational burden on the blockchain, network transmission delays during the signature verification process may affect the overall diagnostic processing speed in a Web3 environment. To mitigate this issue, future research should consider optimization techniques such as parallel processing.

Third, while this study demonstrated the feasibility of applying ZKML to CNN-based models for X-ray image classification, it did not sufficiently explore the applicability of ZKML to other AI architectures such as Transformers, ViT (Vision Transformers), and LSTMs. These models involve complex computational structures, including multi-head attention, patch embeddings, and recurrent layers, which make efficient conversion into ZK circuits challenging. Such structural characteristics can significantly impact proof generation time and memory usage. Therefore, future research should focus on benchmarking various AI architectures and designing customized ZK circuits suited for each model.

5.2 Future Research Directions

This study experimentally validated the applicability of the ZKML framework in a Web3-based medical environment, demonstrating that it can simultaneously ensure privacy protection and verifiability of AI diagnostic results. Based on these experimental results, the following future research directions are proposed.

First, optimization of ZK proof generation speed is essential. While the verification speed was measured at an average of 28.5 ms, the proof generation process still requires high computational resources. Circuit optimization techniques based on ZK-SNARKs, proof size reduction algorithms, lookup tables, and parallel processing structures should be applied to accelerate proof generation. Additionally, adopting ZK-STARKs, which do not require a trusted setup, may enhance security and trust in decentralized environments.

Second, the development of lightweight ZKML models is necessary. Although this study utilized a high-performance GPU (RTX 3090), such

hardware is not readily available in typical medical institutions or edge environments. Therefore, model pruning, quantization, and lightweight circuit design must be explored to build ZKML models that can operate efficiently in low-resource environments, including mobile platforms.

Third, research should be expanded to apply ZKML to various types of medical data and AI models. While this study focused on a CNN model using CXR images, future research should investigate the applicability of ZKML to other medical imaging modalities such as CT and MRI, as well as different AI architectures including Transformers and LSTMs. In particular, integrating FL with ZKML would allow institutions to maintain data privacy while verifying prediction accuracy through zero-knowledge proofs without the need to share raw data.

Finally, further advancement is needed in integrating ZKML with blockchain-based smart contract systems. There is a need to further enhance the integration between ZKML and blockchain-based smart contract systems. In this study, a structure was implemented in which the signature of a hash value recorded on-chain was verified through a smart contract. Building upon this, future work should focus on fully automating the proof verification process through smart contracts and developing a trusted medical data verification framework that includes on-chain model registration, verification, and logging.

These research directions will play a crucial role in establishing ZKML as a core technology within the Web3 medical ecosystem, contributing to the development of the next generation of AI-driven medical systems that ensure both trust and privacy.

5.3 Conclusion

This study presents and experimentally evaluates a framework that enhances the trustworthiness and privacy protection of ML-based medical diagnostics by applying ZKML in a Web3-based medical data environment. A ZK proof generation mechanism was applied to a CNN-based COVID-19 diagnostic model, where the proof generation and verification were conducted off-chain, while the resulting hash value was recorded on-chain and verified through a smart contract. As a result, the proposed ZKML-applied model maintained a high diagnostic accuracy of 88% and achieved an average proof verification speed of 28.5 milliseconds, confirming its applicability in real-time environments. Additionally, the system successfully passed verification tests using both valid and invalid inputs, thereby demonstrating the reliability of

AI predictions through ZK proofs. The entire ZKML implementation process has been made publicly available on GitHub, providing a solid foundation for future research. Despite these achievements, the computational cost of ZK proof generation remains a significant limitation. In particular, the complexity of converting deep learning models into ZK-compatible formats and the computational overhead associated with proof generation may hinder practical deployment in resource-constrained environments.

To overcome these limitations, future research will explore ZK-SNARK-based circuit optimization techniques, the use of lightweight model architecture, and integration with blockchain based medical data verification systems. These efforts aim to improve system scalability and responsiveness while maintaining strong privacy and security guarantees.

This study empirically demonstrates the applicability of ZKML in the medical domain and lays the groundwork for its expansion to a wider range of medical AI models. It contributes to the development of decentralized, privacy-preserving next-generation medical AI systems and suggests that ZKML can serve as a core technology in Web3-based medical environments.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00397538, development of public opinion polling technology based on Web3 that ensures fairness, anonymity, and transparency), 50%.

References

[1] I. D. Apostolopoulos and T. A. Mpesiana, "COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," Physical and Engineering Sciences in Medicine, vol. 43, pp. 635–640, 2020.

[2] E. Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto, "Unveiling COVID-19 from chest X-ray with Deep Learning: A hurdles race with small data," International Journal of Environmental Research and Public Health, vol. 17, no. 18, p. 6933, 2020. https://doi.org/10.3390/ijerph17186933.

[3] M. E. H. Chowdhury, T. Rahman, A. Khandakar, et al., "Can AI help in screening viral and COVID-19 pneumonia?" IEEE Access, vol. 8, pp. 132665–132676, 2020.
[4] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, et al, " Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19," IEEE Reviews in Biomedical Engineering, vol. 14, pp. 4-15, 2020.
[5] S. Pati, S. Kumar, A. Varma, B. Edwards, et al., "Privacy preservation for federated learning in health care," Patterns, vol. 5, 2024.
[6] G. Franklin, R. Stephens, M. Piracha, S. Tiosano, et al., " The sociodemographic biases in Machine Learning algorithms: A biomedical informatics perspective," MDPI, vol. 14, no. 6, 2024.
[7] Z. C. Lipton, " The mythos of model interpretability: In Machine Learning, the concept of interpretability is both important and slippery," ACM Queue, 2018.
[8] B. J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "ZKML: An optimizing system for ML inference in Zero-Knowledge Proofs," in EuroSys '24: Proceedings of the Nineteenth European Conference on Computer Systems, pp. 560–574, 2024.
[9] M. Chinnaiah, A. Gupta, S. Srivastave, et al., "Zero-Knowledge AI: Privacy-first ML inference in distributed ecosystems," in IEEE 5th International Conference on Emerging Research in Electronics, Computer Science and Technology, 2025.
[10] A. D. Santis and G. Persiano, "Zero-Knowledge Proofs of knowledge without interaction," IEEE, 1992.
[11] V. Keršič, S. Karakatič, and M. Turkanović, "On-chain zero-knowledge Machine Learning: An overview and comparison," Journal of King Saud University-Computer and Information Sciences, vol. 36, 2024.
[12] Z. Li, J. Xu, and B. Wang, "Efficient privacy aggregation method based on Zero-Knowledge Proofs in federated learning," in IEEE 2024 7th International Conference on Computer Information Science and Application Technology (CISAT), 2024.
[13] H. Chen, S. U. Hussain, F. Boemer, E. Stapf, et al., "Developing privacy-preserving AI systems: The lessons learned," in IEEE 2020 57th ACM/IEEE Design Automation Conference (DAC), 2020.
[14] B. O. Roelink, M. El-Hajj, and D. Sarmah, "Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication," Security and Privacy, 2024.

[15] H. Lycklama, A. Viand, N. Avramov, et al., "Artemis: Efficient commit-and-prove SNARKs for zkML," arXiv, 2024.
[16] K. Sharifani and M. Amini, "Machine Learning and Deep Learning: A review of methods and applications," World Information Technology and Engineering Journal, vol. 10, pp. 3897–3904, 2023.
[17] A. Alkhalil, A. Razzaq, A. Ahmad, M. Abdelrhman, et al., "A framework for blockchain-based secure management of mobile healthcare (mHealth) systems," Journal of Web Engineering, vol. 24, no. 3, 2025.
[18] X. Song, Z. Wang, K.D. Baek, and K. In-Young, "Personalized user models in a real-world edge computing environment: A peer-to-peer federated learning framework," Journal of Web Engineering, vol. 23, no. 8, 2025.
[19] J. Kim, J. Nang, J. Choe, "LMLT: Low-to-high multi-level vision transformer for lightweight image super-resolution," in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), 2025.
[20] J. Shin, H. Yang, Y. Yi, "SparseInfer: Training-free prediction of activation sparsity for fast LLM inference," in Proc. Design, Automation & Test in Europe Conf., 2025.
[21] D. Hwang, S. J. Oh, J. Choe, "Small object matters in weakly supervised object localization," Neurocomputing, 2025.
[22] M. Lee, K. Song, J. Choe, "Fog-free training for foggy scene understanding," Pattern Recognition Letters, pp. 129–135, 2025.
[23] D. Hwang, H. Kim, D. Baek, H. Kim, I. Kye, J. Choe, "Curriculum learning with class-label composition for weakly supervised semantic segmentation," Pattern Recognition Letters, pp. 171–177, 2025.
[24] COVID-19 Radiography Database https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

Biographies

Jin Ah Seo is a sixth-semester Ph.D. student in the Department of Computer Science at Sogang University. She received her master's degree from the Graduate School of Information and Communication at Sogang University in 2022. Her research interests include blockchain security, with a current focus on Zero-Knowledge Machine Learning (ZKML) and AI security requirements.

Kun Hwa Lee is a second-semester master's degree student in the Department of Electrical and Computer Engineering at Seoul National University. He received his bachelor's degree in Computer Science Engineering from Sogang University in 2025. His research interests are in Machine Learning Security and Privacy. He is currently working on Model IP Protection in Federated Learning.

Vijayan Sugumaran is a Distinguished University Professor and Janke Scholar of Management Information Systems in the School of Business Administration at Oakland University, Rochester, Michigan, USA. He is also the Chair of the Department of Decision and Information Sciences, Co-Director of the Institute for Data Science, and Director of the Master of Science in Business Analytics program. He received his Ph.D. in Information Technology from George Mason University, Fairfax, Virginia, USA. His research interests are in the areas of Big Data Management and Analytics, Ontologies and Semantic Web, Intelligent Agent and Multi-Agent Systems. Sugumaran is the Co-PI on a US$2 million NSF grant to train students in STEM-driven data science and entrepreneurship. He has published over 350 peer-reviewed articles in journals, conferences, and books. He has edited 20 books and serves on the Editorial Board of eight journals. He has published in top-tier journals such as *Information Systems Research, ACM Transactions on Database Systems*, *Communications of the ACM*, *IEEE Transactions on Big Data*, *IEEE Transactions on Engineering Management*, *IEEE Transactions on Education*, *IEEE Transactions on Cybernetics*, *IEEE Multimedia*, and *IEEE Software*. Sugumaran is the editor-in-chief of the *International Journal of Intelligent Information Technologies* and *Journal of Web Engineering*. He is the Chair of the Intelligent Agent and Multi-Agent Systems mini-track for Americas Conference on Information Systems (AMCIS 1999–2025). Sugumaran has served as the Program Chair for the 14th Workshop on E-Business (WeB2015), the International Conference on Applications of Natural Language to Information Systems (NLDB 2008, NLDB 2013, NLDB 2016, NLDB 2019, NLDB 2023, and NLDB 2024), 29th Australasian Conference on Information Systems (ACIS 2018), 14th Annual Conference of Midwest Association for Information Systems (MWAIS 2019), 5th IEEE International Conference on Big Data Service and Applications (BDS 2019), and 2022 Midwest Decision Sciences Institute Annual Conference (MWDSI

2022). He also regularly serves as a program committee member for numerous national and international conferences.

Jo Yeon Park graduated in 2026 from the Department of Computer Science at Sogang University. She received her bachelor's degree in Intellectual Property from Kyonggi University in 2023. Her research interests focus on the convergence of blockchain technology and intellectual property, and she has worked on Zero-Knowledge Machine Learning (ZKML) technologies.

Soo Yong Park received his Ph.D. degree from George Mason University in 1995. He has held several prestigious positions, including serving as a Professor in the Department of Computer Science at Sogang University since March 1998 and as the Dean of the College of Software Convergence at Sogang University since July 2024. He is currently serving as the Chair of the Distributed Ledger Standard Forum and has been the Director of the Web 3.0 Research Center (ITRC) since 2023. Previously, he served as the President and CEO of the National IT Industry Promotion Agency (NIPA) from September 2012 to November 2014. Since January 2019, he has also served

as the President of the Korea Society of Blockchain and as the Director of the Intelligent Blockchain Research Center. His accolades include the 10-Year Most Influential Paper Award from the Asia-Pacific Software Engineering Conference (APSEC) in December 2018 and the Minister of Science and ICT Award for Best Project Evaluation in November 2021.