X-CHAIN: Enhancing Electronic Supply Chain Security with 3D X-ray Inspection and Blockchain Integration

Shuvodip Maitra[1], Tishya Sarma Sarkar[1], Chandan Kumar[1], Abhishek Chakraborty[1], and Debdeep Mukhopadhyay[1]

Indian Institute of Technology Kharagpur, West Bengal, India
shuvodipmaitra@iitkgp.ac.in, tishya@kgpian.iitkgp.ac.in, cchaudhary278@kgpian.iitkgp.ac.in, achakraborty25@kgpian.iitkgp.ac.in, debdeep@cse.iitkgp.ac.in

Abstract. In this study, we address critical vulnerabilities in modern global electronic device supply chains, including risks of counterfeiting, malicious modifications, and hardware Trojan insertion, by proposing **X-CHAIN**, a comprehensive framework that employs a verifier as the root of trust. The verifier leverages X-ray scanning and image processing to classify original and counterfeit printed circuit board (PCB) devices and records the information of each scanned PCB within a blockchain network. The effectiveness of the framework is demonstrated through experiments on eight distinct lightweight and commercial PCBs relevant to Internet of Things (IoT) applications. Beyond X-ray scanning, a two-tier image classification pipeline is employed to determine both the device class and authenticity, followed by localization of suspicious regions in counterfeit devices, using popular deep learning models such as DenseNet-121, Xception, and Inception-V3, achieving classification accuracies exceeding 99%. Furthermore, the proposed blockchain framework is implemented using the Go-Ethereum library, with smart contract deployment gas costs of approximately $5 USD. To the best of our knowledge, this work presents the first unified framework that integrates PCB X-ray imaging with a blockchain-based system to manage and secure the electronic supply chain, wherein X-ray scans authenticate devices and the Ethereum-based blockchain continuously monitors the supply chain from original equipment manufacturers (OEMs) to end users[1].

Keywords: Printed Circuit Board · Counterfeit Detection · Electronic Supply Chain · Blockchain · 3D X-ray Scanning · Image Classification

1 Introduction

Printed Circuit Boards (PCBs) are fundamental to modern electronic systems, yet their highly distributed supply chain—spanning schematic design, layout, fabrication, and assembly across geographically dispersed vendors—introduces

[1] https://github.com/shuvodipmaitra/PCB_Verification_Blockchain

significant risks of malicious modifications, such as microscopic trace alterations or insertion of extra components, that can compromise functionality and security [19]. Visual inspection is often insufficient, as PCBs from trusted and untrusted sources can appear identical, making authenticity verification, Hardware Trojan detection, prevention of design tampering, and mitigation of counterfeiting critically important [9]. X-ray microscope (XRM) based X-ray computed tomography (XCT) has therefore emerged as an effective non-destructive technique for counterfeit PCB detection, using 2D cone-beam X-ray projections reconstructed into high-resolution ($\sim$ 500 nm) 3D volumes via filtered back-projection, enabling automated and rapid analysis [3,25,26]. However, such verification results are typically confined to the verifying authority and are not shared across the supply chain. To address this, blockchain technology offers an immutable, transparent, and tamper-resistant ledger for securely storing and disseminating PCB verification records, while smart contracts enable automated verification and reporting workflows [14]. With declining XRM costs, centralized or government-authorized entities can perform XCT-based certification and record results on a blockchain, allowing customers to query authenticity using a unique PCB device ID before purchase. Accordingly, this work proposes **X-CHAIN**, a framework that integrates deep learning (DL)-based classification of PCB XCT images with blockchain to detect counterfeits, register verification outcomes, and provide a trusted, transparent, and end-to-end record of PCB authenticity for OEMs, vendors, verifiers, and customers throughout the electronics supply chain.

2 Background

X-ray imaging has emerged as a powerful non-destructive tool for the physical assurance of electronic devices such as PCBs and ICs, offering rapid inspection capabilities with real-time 2D projections and 3D XCT scans (which are available within a few hours), while enabling extensive automation through DL-based techniques for classification (AlexNet [2], VGG-Net [24], GAN [39]), object detection (R-CNN + VGG-16 [1], SSD + Inception V2 [16]), and segmentation (D-DOAS [18]), as well as reverse engineering applications [3,26]. The availability of large-scale X-ray image datasets such as SIXray [20] and PIDray [38] facilitates domain-specific transfer learning, yielding superior performance compared to models trained solely on ImageNet and significantly advancing automated PCB inspection. Concurrently, the widespread proliferation of ICs and PCBs has heightened vulnerabilities to counterfeiting, piracy, and hardware Trojans, motivating mitigation approaches including hardware metering [15], IC camouflaging [22], split manufacturing [23], hardware watermarking [5], and physically unclonable functions (PUFs) [29]. Blockchain, originally introduced for cryptocurrency systems, provides an immutable and decentralized framework wherein stakeholders across the PCB supply chain—such as OEMs, foundries, assemblers, IP owners, distributors, retailers, and vendors—can enhance transparency and trust, and has been applied to prevent remarking, recycling, and

cloning [35]. Recent studies have also validated PUF-based PCB authentication for blockchain integration [6] and demonstrated the detection of recycled ICs using confidence-level modeling of CDIR sensor data within a blockchain [32].

3 Our contributions

In summary, we broadly divide our contributions into three folds:

- **Dataset Generation:** We collect 2D reconstructed image slices of 8 different PCB boards, namely Arduino Uno, Raspberry Pi-4, FRDM-KL25Z MCU development platform, Cora Z7 FPGA development board, Asus Tinker board, STM32F4 Discovery Kit, CC3220SF LaunchPad, and Basys-3 FPGA development board, utilizing our in-house XRM facility. The sample's projection images are acquired by rotating them to different angles ranging from 0° to 360°.
- **Deep Learning based classification:** We propose a two-level DL-based classification technique to classify the PCB devices. We use the reconstructed slices from each board to train Inception-V3, DenseNet-121, and Xception models in the first level of classification with a validation accuracy of 99.48%. Later, in the second level classification, we utilize an unsupervised learning algorithm such as a one-class support vector machine (SVM), an isolation forest, and an autoencoder to determine the authenticity of the classified PCB. We also augment the dataset using adversarial and noise addition-based training (to account for a scenario where there is scarcity in acquiring physical duplicate PCB samples), and train the one-class SVM model on this augmented dataset. Lastly, we use a gradient-based class activation mapping (Grad-CAM) technique to identify suspicious areas within counterfeit boards in the form of heatmap images.
- **Integrating with Blockchain Framework:** We are the first to utilize DL-based classification results, as well as the aforementioned heatmap images, as a token of information to store on the blockchain network for verification purposes. We design a corresponding smart contract equipped with functionalities like registering different nodes and devices present as a part of the supply chain network. The minimal cost of deploying our smart contract on the Go-Ethereum blockchain network shows the feasibility of our framework. We also show that this type of Blockchain can be implemented in a post-quantum secure manner, using Zama's FHEVM framework. This ensures privacy and integrity of data in the electronics supply chain, while executing the necessary functionalities on encrypted data.

4 Threat Model and Verification-based defense setup

Based on the pervasive threat of counterfeit devices in the PCB supply chain, this section presents a comprehensive threat model along with a plausible defense mechanism. The adversary is assumed to be a malicious entity—such as

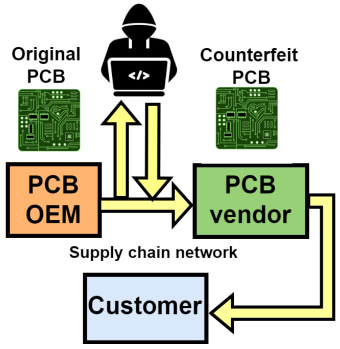


Fig. 1: Threat Model

a rival company, counterfeiters, or rogue state and non-state actors—seeking to disrupt the PCB supply chain between OEMs and vendors by gaining access to original PCBs during transit through international and national distributors, and subsequently fabricating cloned devices using these originals as references. These counterfeit PCBs are physically indistinguishable from genuine ones and are reintroduced into the supply chain masquerading as authentic products, as illustrated in Fig. 1. The adversary's motivations include financial gain through low-cost counterfeit sales, market sabotage by introducing subtle defects that cause premature failures, and security breaches via hardware Trojans targeting critical infrastructures such as power grids, data centers, and military systems. Such compromises may involve adding or removing components, substituting substandard parts, altering the branding, tampering with copper traces or vias to intercept or disrupt signals, injecting malicious firmware or software, modifying electrical parameters, exploiting diagnostic test points, or concealing Trojans within hidden or altered PCB layers. These methods undermine device confidentiality, integrity, reliability, and functionality. As a defense, verification is performed by certified government laboratories or law enforcement agencies equipped with state-of-the-art XRM systems to acquire high-resolution XCT scans, enabling the capture of micron-scale physical variations. These XCT scans are processed on GPU workstations where DL–based feature extraction and classification methods accurately distinguish original PCBs from duplicates.

5 Proposed Framework

In this section, we describe our proposed methodology, which can be broadly divided into three parts: 1) Image acquisition, 2) DL-based classification to classify the PCB type, detect counterfeit PCBs, and identify suspicious areas within counterfeit PCBs; and 3) Blockchain-based framework to integrate DL-based results with the supply chain network.

5.1 Image Acquisition

Our setup includes an XRM instrument (Zeiss Xradia-515) that is used for generating XCT scans of PCB samples. The detailed specifications of the XRM model are discussed in the Appendix A. For our experiments with different PCB boards, we have a source voltage of 80 kV, power of 7 W, detector objective lens $0.4X$, and voxel size of around 50 µm- 52 µm. A full session scan renders a 3D reconstructed model of the PCB sample. This 3D reconstructed image is composed of a stack of 2D reconstructed image slices (top to bottom), which form the XCT scans of the PCB sample. We perform state-of-the-art image processing analysis on these 2D reconstructed slices. The XCT scans of our PCB samples, acquired through the setup described here, serve as the foundational dataset for our deep learning-based framework, which is discussed in the following section.

5.2 Deep Learning based Framework

In this section, we discuss the DL-based analysis of the PCB XCT dataset. We use a two-tier classification system to generate two distinct tokens, which are utilized within the blockchain network. Then we perform a Grad-CAM analysis to identify 3D heatmap images, which are flattened and converted into a single token. Thereafter, all three tokens are fed into the blockchain network. Before the training phase, various data processing techniques are employed to aid in the extraction of features using DL models. These are briefly discussed in the following subsections.

5.2.1 Base level classification: The base-level supervised classification is performed first to determine the type or family of the PCB device. We have 8988 XCT images divided into 8 separate classes. The distribution of images are: Arduino Uno (1008 images), Raspberry Pi-4 (1264 images), FRDM-KL25Z MCU development platform (924 images), Cora Z7 FPGA development board (924 images), Asus Tinker board (1424 images), STM32F4 Discovery Kit (840 images), CC3220SF LaunchPad (1176 images), and Basys-3 FPGA development board (1176 images). These images are labeled using one-hot encoding before the training. The data is divided into training and validation sets using the standard $80:20$ rule. This prevents overfitting of the model. Our optimized CNN model captures the image features using transfer learning. We compare three state-of-the-art CNN architectures namely, Inception-V3 [30], DenseNet-121 [10], and Xception [7]. The top layers of the base models are modified, where three dense layers with 256, 128 and 64 neurons each are stacked along with dropout layers in between. We use the rectified linear unit (ReLU) activation function, which succinctly reduces the computational complexity of the neural networks. The final classifier layer consists of 8 nodes, with softmax activation. We use an Adam optimizer for model training. Adam [13] optimizer dynamically adjusts the learning rate for each parameter during training based on past gradients and updates accordingly. This allows faster convergence and is more reliable compared to fixed learning rate methods, especially when dealing with sparse gradients or

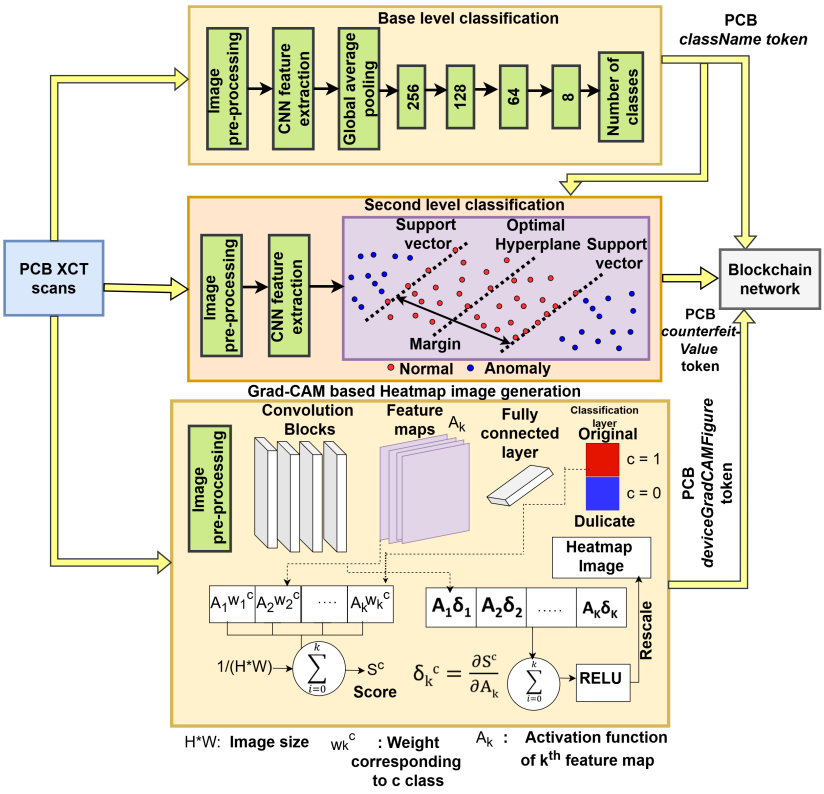


Fig. 2: Image Processing and DL-based Classification Flow

non-stationary objectives commonly encountered in CNN training. We experiment with different batch sizes, number of epochs, learning rates, and number of dense layers to obtain the optimum accuracy and loss values. The output of this stage is fed to our proposed blockchain network during PCB device registration as the *className* token. The DL based image processing flow of PCB XCT scans is depicted in Fig. 2.

5.2.2 Second level classification: After the base classification, a second-level classification is performed to determine whether the PCB is original or counterfeit. There are two assumptions. The **first case assumption** is that there are many original and duplicate PCB boards available with the verifier during the training process. The **second case assumption** is that the verifier has no access to the duplicate boards and has only the original PCB of each class, acquired directly from the PCB OEMs.

For the **first case assumption**, we consider a dataset of 19 Arduino Uno boards (as a reference), where there are 9 original Arduino Uno boards containing 3501 XCT, and 10 duplicate Arduino Uno boards (obtained from different ven-

dors) containing 3890 XCT images. On this dataset, we perform unsupervised classification using one-class SVM, isolation forest, and a CNN-based autoencoder. Here, the machine learning model trains on the original images and then predicts on the duplicate images. The models are trained on feature representations extracted exclusively from the original images, enabling the model to learn the distribution of normal data. DenseNet-121, with its classifier removed, is used as a pre-trained feature extractor to convert each X-ray scan into a 50,176-dimensional embedding, thereby ensuring that our machine-learning models operate on rich, semantically meaningful descriptors rather than raw pixels. The one-class SVM is configured with an RBF kernel, which allows the model to learn a flexible non-linear boundary enclosing the normal samples. During inference, predictions yield $+1$ for normal/original instances and -1 for anomalies. The isolation forest model learns typical feature-space structures through random partitioning trees. Each tree isolates samples by recursively performing random splits, and anomalies—such as duplicate images—tend to require fewer splits to isolate. A CNN-based deep autoencoder is being trained using feature vectors extracted from original images to learn a compact representation of normal behavior. DenseNet-121 is used as a fixed feature extractor, with a 512-dimensional latent space, and then reconstructed. During inference, duplicate images are processed through the autoencoder, and their reconstruction errors are evaluated against an anomaly threshold defined as the mean (μ) training set reconstruction error plus two times the standard deviation (σ).

For the **second case assumption**, the verifier has access to only original images and trains its model by artificially augmenting the limited dataset by adversarial learning and noise addition. In this scenario, we choose a single original Arduino Uno, and a single duplicate Arduino Uno board (1008 X-ray scan images each) and perform both adversarial perturbation and noise addition on the original images to mimic the variations present in the duplicates. We used the one-class SVM model to learn the features from the original images and then perform prediction on the duplicate images. The one-class SVM was trained solely on the feature distribution of the augmented original samples, which includes original images and their adversarial and noisy variants. This type of training makes the model more robust to the chances of misclassifications.

Adversarial training is a defense mechanism that enhances model robustness by jointly training on clean and adversarially perturbed samples, with the objective of minimizing the worst-case loss under a bounded perturbation ϵ, formulated as $\min_\theta \ \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\|\delta\|_\infty\leq\epsilon} J(\theta, x+\delta, y)\right]$, where $J(\theta, x, y)$ denotes the loss function, δ the adversarial perturbation, and θ the model parameters [33]. A widely used method for generating such perturbations is the Fast Gradient Sign Method (FGSM), which constructs adversarial examples by perturbing the input in the direction of the sign of the loss gradient, given by $x^{\text{adv}} = x + \epsilon \cdot \text{sign}\left(\nabla_x J(\theta, x, y)\right)$, where ϵ controls the perturbation magnitude and hence the attack strength and perceptibility [28, 33]. Training with FGSM-generated samples improves robustness against gradient-based attacks; however, stronger defenses are achieved using Projected Gradient Descent (PGD), an it-

erative extension of FGSM that applies multiple small perturbation steps while projecting the adversarial example onto an ℓ_∞-ball of radius ϵ, using the update rule $x^{t+1} = \Pi_{B_\epsilon(x)}\left(x^t + \alpha \cdot \text{sign}\left(\nabla_x J(\theta, x^t, y)\right)\right)$, where α is the step size and $\Pi_{B_\epsilon(x)}(\cdot)$ enforces the constraint $\|x^{t+1} - x\|_\infty \leq \epsilon$ [33,34]. PGD adversarial training, by incorporating these iteratively generated adversarial examples, is widely regarded as providing stronger and more reliable robustness compared to FGSM-based approaches.

Noise addition is a widely used technique to simulate real-world distortions and assess model robustness in machine learning and computer vision [31]. Gaussian noise corrupts an input x by adding random samples drawn from a normal distribution, yielding a noisy input $\tilde{x} = x + \mathcal{N}(0, \sigma^2)$, where $\mathcal{N}(0, \sigma^2)$ has zero mean and variance σ^2, and the standard deviation σ controls the noise intensity, ranging from mild perturbations to severe corruption that can obscure important features [8]. In contrast, Salt-and-Pepper noise introduces sparse, high-intensity distortions by randomly setting individual pixels x_i to either the minimum (pepper) or maximum (salt) value, such that $\tilde{x}_i = 0$ with probability p_{pepper}, $\tilde{x}_i = 1$ with probability p_{salt}, and $\tilde{x}_i = x_i$ with probability $1 - (p_{salt} + p_{pepper})$, where larger values of p_{salt} and p_{pepper} increase the proportion of corrupted pixels and provide a stringent test of model resilience to sparse impulse noise [11].

The output of the second level classification stage is fed to our proposed blockchain network during PCB device registration as *counterfeitValue* token, which is a binary variable where 1 means the PCB device is a counterfeit one, and 0 means it is an original PCB device.

5.2.3 Grad-CAM based visualization: Grad-CAM (Gradient-weighted Class Activation Mapping) is an explainability technique used to visualize and interpret the decision-making process of convolutional neural networks (CNNs) by producing class-discriminative localization maps that highlight image regions most relevant to a given prediction [17, 27]. It operates by computing the gradients of the class score (before softmax) with respect to the feature map activations of the final convolutional layer, globally averaging these gradients to obtain neuron importance weights, and forming a weighted combination of the forward activation maps followed by a ReLU operation to emphasize only positively contributing features. In the considered setup using DenseNet-121, let there be k feature maps extracted from the input image and passed to a fully connected layer corresponding to $c = 2$ classes (original and duplicate); the class score S_c is computed as $S_c = \frac{1}{H \times W} \times \Sigma_k(w_k^c \times A_k)$, where A_k denotes the activation of the k^{th} feature map, w_k^c its weight contribution to class c, and H and W are the spatial dimensions of the feature maps. The gradients $\delta_k^c = \frac{\partial S_c}{\partial A_k}$ are then computed and combined with the activations to generate the Grad-CAM heatmap as $H^c = \text{ReLU}(\Sigma_k(\delta_k^c \times A_k))$, where ReLU ensures that only features with a positive influence on class c are retained, leading to improved localization. In this work, Grad-CAM is applied in conjunction with a one-class SVM using DenseNet-121 as a feature extractor to visualize regions influencing the classification of images as original or potentially counterfeit, and the resulting RGB

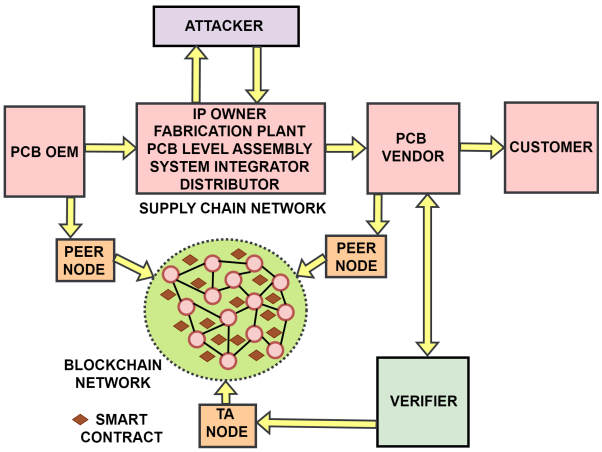


Fig. 3: Local-level Blockchain framework

heatmap is flattened into its corresponding bits and stored as a *deviceFigure* token that is subsequently transmitted to the blockchain.

5.3 Blockchain based Framework

The tokens generated by the deep learning framework are integrated into our proposed blockchain network during PCB device registration. The transparency of the blockchain network improves visibility into the movement of PCBs across various stages of production, shipping, and delivery, enabling better traceability and accountability. Lastly, blockchain maintains a complete and auditable record of all transactions.

An Overview of the Local-level Blockchain network: This section describes the role of the verifier in monitoring the electronics supply chain. The verifier stores ML-based test results for each PCB in a secure blockchain framework, as shown in Fig. 3. The verifier operates as a trusted authority (TA) node, while PCB OEMs and vendors act as peer nodes. OEMs enroll with the TA and register new PCB classes, after which the TA acquires sample boards, performs XCT scans, and trains DL-based models for classification. Once trained, the model is fixed.

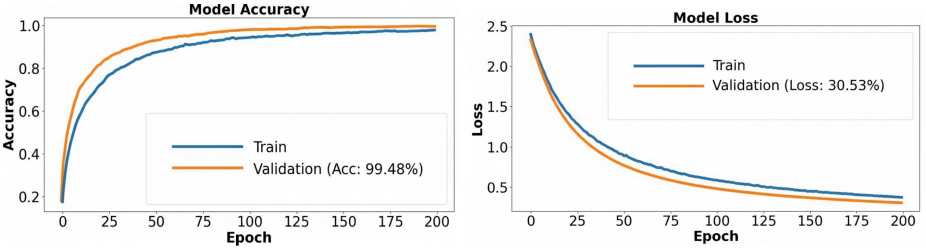
Registered vendors notify the blockchain upon receiving new PCB devices. The TA then acquires these devices, performs XCT scanning, and uses the trained model to identify the PCB class, determine authenticity, and generate associated metadata. Each device is registered on the blockchain with a unique ID, ownership and verifier addresses, classification results, and pricing information, after which the device is returned to the vendor. All device records are

immutably stored on the blockchain and made publicly accessible. Ownership transfer and payment execution are enforced through smart contracts, ensuring that payment is released to the seller only if the PCB is verified as authentic.

Overview of the Global-level Blockchain Network: Since the local-level blockchain (described previously) involves physical acquisition and imaging-based characterization of individual PCB devices, it is difficult to scale it up over a large geographical area. Therefore, we propose that there be multiple TA nodes spread across a large area. Each TA node is responsible for administering a local area. Within this local area of responsibility, the TA node is empowered to register new OEMs, vendors, and PCB devices. The Blockchain is deployed with a few initial TA nodes. The master TA node acts as the chief administrator of the blockchain. The seniormost TA node by default becomes the master TA node initially. Thereafter, each TA node is empowered to cast its vote in favor of its choice of master TA node through an electronic voting process to determine the new master TA node. The winning candidate is determined using a majority vote. If one or more TA nodes get the same number of votes, then the winner is determined by seniority. The master TA node is responsible for registering new TA nodes. The master TA node keeps track of the number of duplicate PCB devices (where counterfeit value is 1) available with each of the PCB vendors. If any PCB vendor has more than 5 counterfeit boards (of any ($className$)), then this PCB vendor node is revoked by the master TA node.

6 Results and Discussion

In this section, we discuss in detail the results that we acquire at each level of verification of the PCB samples. There are a few primary metrics that dictate the performance of the DL-based models. Such metrics are training and validation accuracies, precision, recall, F1-score, and area under the receiver operating curve (AUC-ROC). These metrics quantify the efficacy of the DL models trained on our X-ray CT image dataset. The training and validation accuracies provide a high-level insight into the performance of the DL models. While the training accuracy defines the model's ability to classify the training instances according to their inherent classes, the validation accuracy verifies the same with a separate set of validation instances. Precision is also called the positive predictive value, and it quantifies the true positive (TP) cases out of all predicted positive cases (sum of true positives and false positives). Recall, also known as the true positive rate (TPR) or sensitivity, is the metric that assesses the fraction of true positive predictions relative to all actual positive cases. F1-score represents the harmonic mean of the precision and recall and is particularly important for imbalanced datasets. The AUC-ROC score is commonly applied in binary classification tasks, and it signifies the model's proficiency in distinguishing between positive and negative instances. A higher AUC-ROC score, nearing 1, typically indicates superior discrimination capability between the classes. All the metrics



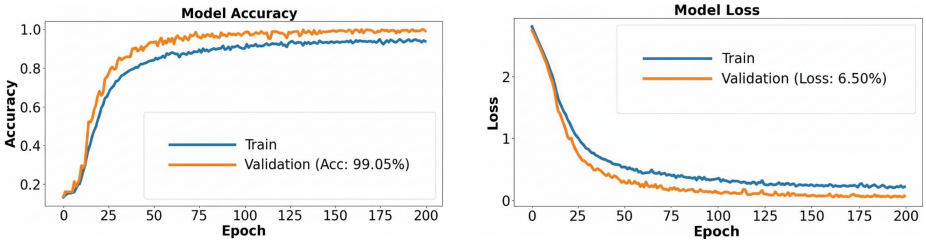
(a) Model Accuracy (b) Model Loss

Fig. 4: Base level Classification Results of Inception-V3.

mentioned above have values ranging from 0 to 1. The detailed results of our DL-based analysis are presented in the following subsections.

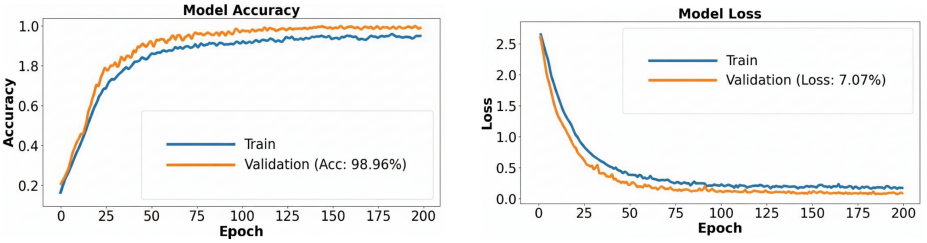
6.1 Base Level Classification Results

The base level classification of our proposed framework determines the family of the individual PCB samples. We employ multiclass classification using three state-of-the-art CNN architectures, leveraging transfer learning. Data augmentation methods improve the performance of the CNN models. Fig. 4, Fig. 5, and Fig. 6 summarize the accuracy and loss values of the base level classification. The Inception-V3 model stands out as the best classifier, boasting a validation accuracy of 99.48%, alongside other peak performance metrics. Both the accuracy and the loss parameters are consistently improving across the epochs. The loss curve converges at 30.53%. Furthermore, the learning curves exhibit notably smoother gradients compared to those of other contemporary CNN models, as evident from Fig. 4a and 4b. The superiority of Inception-V3 over DenseNet-121 and Xception can be attributed to several factors, including the inherent architecture design. The architecture of Inception-V3 is based on the inception module, which integrates multiple branches within each layer. This design facilitates the extraction of a wide range of features in an efficient manner. On the other hand, both the DenseNet-121 and Xception models yield equivalent accuracies along with the other parameters. The validation accuracy achieved for DenseNet-121 is 99.05% (refer Fig. 5a), while for Xception, it reaches 98.96% (refer Fig. 6a).The respective losses of both models converge to 6.50% (refer Fig. 5b) and 7.07% (refer Fig. 6b), respectively. DenseNet-121 employs densely connected blocks, which lead to feature reuse and enhanced gradient flow. Xception, on the other hand, uses depthwise separable convolutions to capture spatial and channel-wise correlations separately, which are effective but require more data for training. Nonetheless, the undulations in the accuracy curves of DenseNet-121 and Xception compared to Inception-V3 could stem from differences in model architecture, training dynamics, and hyperparameter settings. These factors collectively contribute to the observed variations in the convergence behavior of different models during training.



(a) Model Accuracy (b) Model Loss

Fig. 5: Base level Classification Results of DenseNet-121.



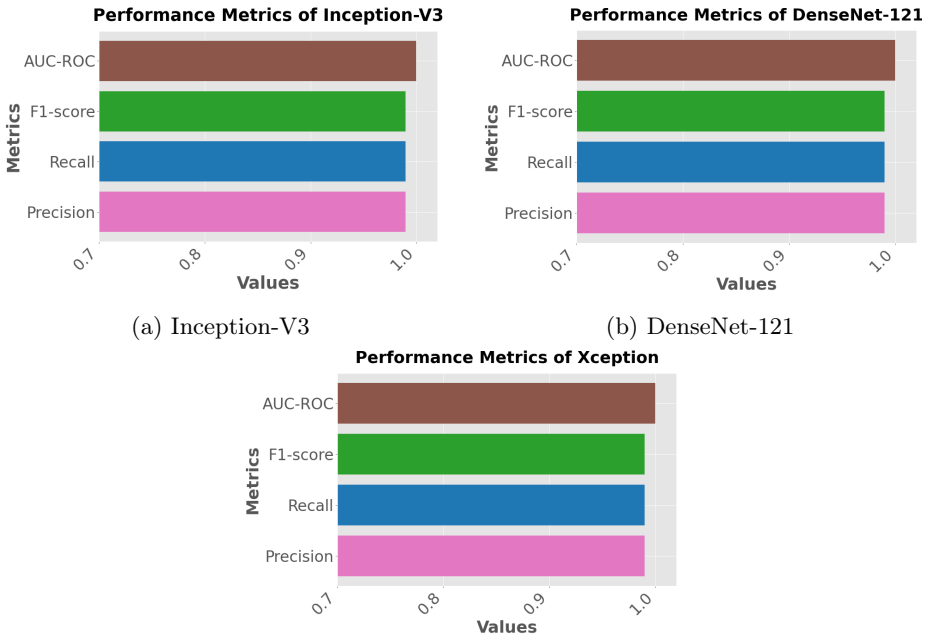
(a) Model Accuracy (b) Model Loss

Fig. 6: Base level Classification Results of XceptionNet.

Table 1 enumerates the accuracy values obtained in different state-of-the-art works related to XCT scan image classification. In comparison to the other contemporary works, our models, especially Inception-V3, perform exceptionally well. To further assess the classification abilities of the DL models, we record the precision, recall, F1-score, and AUC-ROC values of the three models during training. Fig. 7 illustrates the comparative values of precision, recall, F1-score, and the mean AUC-ROC values across the three models, corresponding to the 8 classes of PCBs. The precision, recall, and F1-score for all the models reach a value of 0.99. The high precision value indicates that the models have a low false positive rate. Similarly, a high recall value indicates a strong ability to detect instances that are falsely classified as negative. Additionally, a high F1-score indicates a good balance between precision and recall. We demonstrate

Table 1: Comparison of classification accuracy between our work and state-of-the-art approaches (base layer classification)

Sl. No.	Techniques used	Dataset	Number of images	Accuracy
1.	AlexNet [2]	Private	6997	0.99
2.	VGG [24]	Private	120000	0.995
3.	ResNet-50 + CHR [20]	SixRay10	98219	0.779
4.	GAN + KNN [36]	Private	10000	0.9837
5.	**Our work (Inception-V3)**	**Private**	**8988**	**0.9948**



(a) Inception-V3

(b) DenseNet-121

(c) Xception

Fig. 7: Performance metrics of the three CNN models trained on our PCB x-ray CT dataset.

the mean AUC-ROC scores yielded from the base level classification for all the 8 classes of PCBs. A higher score of AUC-ROC indicates that the model exhibits superior discrimination capability, suggesting that the model achieves higher true positive rates and lower false positive rates across various thresholds. In the realm of multiclass classification, it indicates that the model's predictions for each class are well-calibrated, with higher probabilities assigned to the correct class compared to incorrect ones. Our analysis of the augmented dataset obtains 100% AUC-ROC scores against each PCB class. Thus, our proposed framework reliably generates the accurate *className* token, which is further utilized by the blockchain network. Next, we discuss the results obtained from the second-level classification, which certifies the authenticity of the PCB devices.

6.2 Second Level Classification Results

As discussed in Section 5.2, the second level classification determines the *counterfeitValue* token of the PCB samples. This is a necessary step to prevent any attacker from infecting the supply chain with counterfeit PCBs. For the **first case assumption** (defined in Section 5.2), we first perform the one-class SVM. The key contamination parameter ν (nu) controls the trade-off between the tightness of the decision boundary and the proportion of training points allowed as

outliers. Larger ν results in a looser boundary, making the model more sensitive to deviations. We have tuned the ν value using a grid search and trained our model. The results are presented in Fig. 8, which corresponds to the dependency of the model's precision, recall, and F1 score on ν. The results in the figure clearly illustrate this behavior: at high ν values (0.95–0.9), the model maintains perfect precision (1.0), meaning that all detections are true anomalies, while recall decreases gradually as ν is reduced. As ν becomes smaller (0.8 to 0.6), the SVM boundary tightens, reducing its ability to detect all duplicates, and recall and F1 score drop accordingly. Overall, $\nu = 0.95$ provides the best balance, achieving high recall (0.9374) and the strongest F1 score (0.9662), indicating that a more permissive boundary generalizes better for distinguishing duplicate images from originals in this feature space.

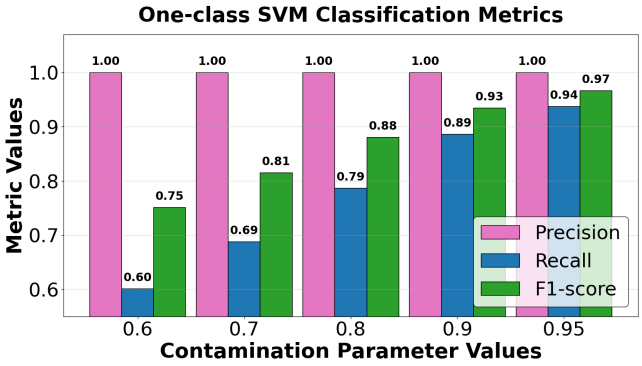


Fig. 8: One-class SVM classification metrics. We vary the contamination parameter, ν and record the corresponding metrics, such as precision, recall, and F1-score for each ν value.

Next, we perform the isolation forest experiments. Several key hyperparameters are varied: $n\_estimators$, the number of trees that improves stability; $max\_samples$, the number of feature vectors used per tree; and $max\_features$, the proportion of features sampled for each split. The results (illustrated in Table 2) show that across all settings, we obtain perfect precision (1.0), meaning the model never falsely labels original-like samples as duplicates, but recall and F1 score vary between 0.6 and 0.7, respectively, depending on the configuration. The best performance occurs at $n\_estimators = 500$ and $max\_features = 0.75$, yielding a recall of 0.6329 and an F1 score of 0.7751. When compared to the One-Class SVM, Isolation Forest performs noticeably worse: the SVM achieved an F1 score of 0.9662 at $\nu = 0.95$, with recall above 0.93, indicating a substantially superior ability to detect all duplicates. While Isolation Forest remains conservative and highly precise, its lower recall confirms that the One-Class SVM is the stronger anomaly detection method for this dataset.

Table 2: Isolation Forest Training

n_estimators (no. of trees)	**max_sample/tree**	**max_features**	**Precision**	**Recall**	**F1 Score**
200	512	0.5	1.0	0.6203	0.7656
		0.75	1.0	0.6221	0.7670
		1.0	1.0	0.5966	0.7473
300	512	0.5	1.0	0.6156	0.7621
		0.75	1.0	0.6269	0.7707
		1.0	1.0	0.6059	0.7546
400	512	0.5	1.0	0.6182	0.7640
		0.75	1.0	0.6244	0.7687
		1.0	1.0	0.6239	0.7684
500	512	0.5	1.0	0.6187	0.7644
		0.75	1.0	0.6329	0.7751
		1.0	1.0	0.6143	0.7611

Thereafter, autoencoder-based anomaly detection experiments are performed using DenseNet-121 feature vectors as inputs to a fully connected autoencoder with a 2048–1024–512 encoder and a symmetric decoder. The autoencoder compresses high-dimensional embeddings into a 512-dimensional latent space before reconstruction. The autoencoder is trained for 50 epochs using the Adam optimizer to minimize the mean squared reconstruction error. During inference, duplicate images are identified by comparing their reconstruction errors against an anomaly threshold defined as the mean training reconstruction error plus two times the standard deviation. Key hyperparameters, including learning rate and batch size, are tuned due to their strong influence on reconstruction sharpness and training stability, with higher learning rates (e.g., 0.001) yielding lower reconstruction errors and clearer separation between original and duplicate samples. As shown in the Table 3, the best configuration (learning rate = 0.001, batch size = 16) achieves a precision of 1.0, a recall of 0.9287, and an F1 score of 0.9630, demonstrating the autoencoder's ability to reliably capture deviations introduced by duplicate images. Compared to the One-Class SVM, which attains an F1 score of 0.9662 at $\nu = 0.95$, the autoencoder exhibits comparable performance with perfect precision but slightly reduced recall, indicating that while both methods effectively detect anomalies in the DenseNet-121 feature space, the One-Class SVM more tightly encloses the normal data distribution, whereas the autoencoder occasionally under-reconstructs borderline duplicate samples, leading to minor recall degradation. Nevertheless, the autoencoder remains a competitive alternative, particularly when richer learned representations are advantageous over boundary-based approaches.

For the **second case assumption** (as defined in Section 5.2), we first perform adversarial learning experiments. In adversarial learning, the parameter ϵ plays a crucial role in the FGSM experiments by controlling the maximum perturbation added to the input image. A small ϵ produces subtle perturbations

Table 3: Autoencoder Training

Learning rate	batch size	Reconstruction error (training)	Anomaly Threshold	Reconstruction error (testing)	Precision	Recall	F1 Score
0.001	16	$\mu = 0.0069$ $\sigma = 0.0034$	0.0128	$\mu = 0.0549$ $\sigma = 0.0558$	1.0	0.9287	0.9630
	32	$\mu = 0.0075$ $\sigma = 0.0039$	0.01518	$\mu = 0.0504$ $\sigma = 0.0502$	1.0	0.8380	0.9118
	64	$\mu = 0.0102$ $\sigma = 0.0053$	0.0208	$\mu = 0.0504$ $\sigma = 0.0526$	1.0	0.7123	0.8320
0.0001	16	$\mu = 0.0066$ $\sigma = 0.0034$	0.013	$\mu = 0.0398$ $\sigma = 0.0401$	1.0	0.7820	0.8776
	32	$\mu = 0.0095$ $\sigma = 0.0049$	0.0193	$\mu = 0.0419$ $\sigma = 0.0419$	1.0	0.6809	0.8102
	64	$\mu = 0.0135$ $\sigma = 0.0067$	0.0269	$\mu = 0.0449$ $\sigma = 0.0426$	1.0	0.5922	0.7439
0.00001	16	$\mu = 0.0184$ $\sigma = 0.0093$	0.037	$\mu = 0.0462$ $\sigma = 0.0431$	1.0	0.4406	0.6117
	32	$\mu = 0.0231$ $\sigma = 0.0112$	0.0455	$\mu = 0.0503$ $\sigma = 0.0470$	1.0	0.3701	0.5403
	64	$\mu = 0.0293$ $\sigma = 0.0142$	0.0577	$\mu = 0.0568$ $\sigma = 0.0531$	1.0	0.3005	0.4621

while a larger ϵ increases the attack strength but may distort the image significantly. For PGD experiments, both ϵ and the step size α influence the attack, where ϵ sets the perturbation budget and α determines the magnitude of update in each iteration, thereby balancing convergence speed and attack effectiveness. In Gaussian noise addition, the standard deviation (σ) defines the spread of the noise distribution, with higher values introducing stronger randomness and potentially overwhelming the image structure. Similarly, in salt-and-pepper noise, the parameters *salt_probability* and *pepper_probability* control the fraction of pixels set to maximum (salt) and minimum (pepper) intensity values, respectively, thereby dictating the density and severity of the noise in the image. Individual effects of the adversarial and noise addition are presented in the Appendix E. The combined effect of adversarial and noise addition on SVM training is depicted in Fig. 9. For the combined adverarial and noise training, we have taken FGSM ($\epsilon = 0.5$); PGD ($\epsilon = 0.5$); (α=0.05); Gaussian noise ($\sigma^2 = 30$) (for raw pixel values); and salt and pepper probability (*salt_probability* = 0.05; and *pepper_probability* = 0.05).

6.3 Grad-CAM based Visualization

For one-class SVM using DenseNet-121 as a feature extractor, we perform the grad-CAM technique to obtain the heatmap for each individual reconstructed slice. Since the last convolution feature map of DenseNet-121 has a shape of

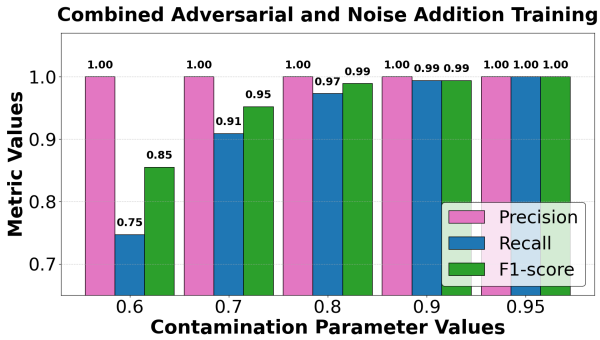


Fig. 9: Combined adversarial and noise addition training

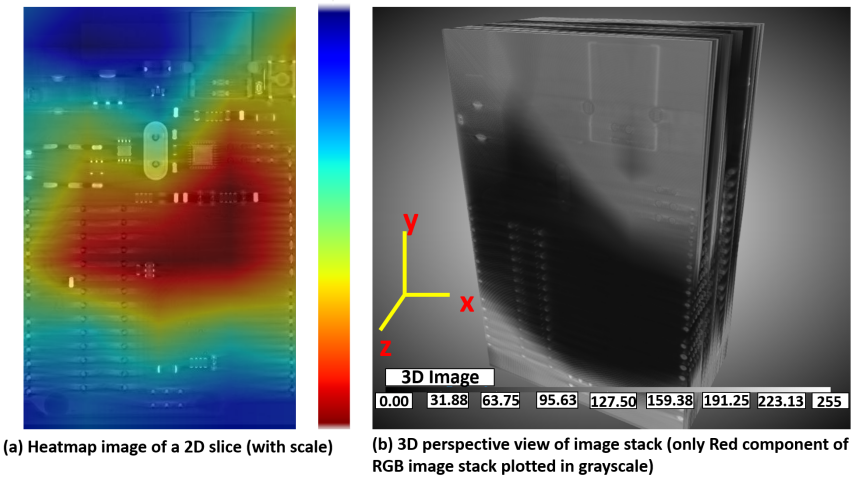


Fig. 10: (a) Generated Heatmap image from the grad-CAM process; (b) Its 3D perspective

(7×7), the heatmaps generated for each of the reconstructed slices have a (7×7) shape. For accurate visualization, these heatmaps are upgraded to the original image resolutions (731×731) pixels and superimposed over the original reconstructed images. In these superimposed images, we can accurately visualize the areas that have a stronger influence on the anomaly probability (highlighted in red) and areas that have minimal influence on the final anomaly probability (highlighted in deep blue), with yellow and green areas in between. In the context of anomaly detection, the red areas highlight the suspicious areas present within the 3D volume of a duplicate PCB. The superimposed images are cropped to allow only the PCB portion, which is our Region of Interest (ROI), negating any influence the background has over the resulting heatmaps. The final image has a resolution of (398×616) pixels. Thus each PCB sample has a 3D heatmap

dataset of $(398 \times 616 \times 252)$, where the last dimension represents the number of reconstructed 2D slices that contain the most relevant information. The resulting 3D heatmap is shown in Fig.10. This 3D heatmap allows us to track the spatial evolution of the suspected regions throughout a sample PCB's volume. Within the heatmap, the red areas are the regions that the model focuses strongly on when making its decision. These correspond to the areas within the PCB that are suspicious from a counterfeiting perspective. Blue areas are the least influential for the model's decision. Green and yellow represent intermediate levels of importance. We further depict the 3D perspective view of the red component of the RGB image stack, in which the majority of the features of the PCB are found. This is represented as a grayscale plot. This 3D perspective image shows the progress of the suspicious areas (red areas of the heatmap obtained from the grad-CAM process), through the depth of the PCB sample.

6.4 Blockchain deployment results

We chose go-ethereum (geth) [4] version 1.13.14-stable-2bd6bd01 and Ganache version 2.7.1 for creating a local Ethereum blockchain network running on Intel® Core™ i5-9500 CPU at 3.00GHz clock frequency with 20 GB RAM and 6 cores. We wrote the smart contract in Solidity and compiled it using the solc v0.8.1 compiler. We use *solc –bin PCBVerification.sol* and *solc –abi PCBVerification.sol* to generate bytecode and abi, respectively. Next, we use a Web3-based Python interface to deploy the contract over the local blockchain. The details of the smart contract named `PCBVerifier` contract are presented in Appendix F. The price of ether fluctuates in the open market, often denominated in USD or other fiat currencies. Gas is the unit used to measure the computational effort required to execute operations on the Ethereum network. Gas costs for smart contract deployment depend on factors like contract complexity and network congestion. Higher gas costs result in higher fees for deploying smart contracts. To estimate the USD cost of deploying a smart contract on Ethereum, one must consider the current exchange rate between Ether and USD. Gas costs are denominated in ether, so the total USD cost depends on both the gas cost and the current ether price. Deploying smart contracts during periods of low network congestion can reduce gas costs. Gas cost (in ETH) = Execution Cost(in gas terms) $\times$ gas rate (price of 1 gas in Gwei). The current gas rate is 0.388 Gwei, and 1 ETH = 4178.73 USD. Based on these values, the costs associated with **Contract deployed** operations and functions like are calculated and presented in Table 4. The table shows that smart contract deployment requires a small one-time investment of about 5 USD. Further, after the deployment of the contract, the execution of functions like **registerOEM**, **registerVendor**, and **registerDevice** is also a one-time investment that ensures a low-cost supply chain transparency oversight. The calculated costs provide insight into the economic feasibility of implementing blockchain-based solutions for electronics supply chain management. With relatively modest initial investment requirements, electronics businesses (both OEMs and vendors) can leverage smart contracts on the Ethereum network to streamline processes and enhance transparency

Table 4: Smart contract Ethereum deployment gas prices (where 1 gas = 0.388 Gwei; 1 ETH = 4178.73 USD)

Operation	Transaction cost (gas)	Execution cost (gas)	Gas price (Gwei)	Gas Price (ETH)	Gas Price (USD)
Contract deployed	3435894	3114780	1208535	0.001209	5.05014
castVote	39677	17877	6936.276	6.94E-06	0.028985
registerOEM	70640	49208	19092.7	1.91E-05	0.079783
registerPCBClass	138788	117548	45608.62	4.56E-05	0.190586
registerVendor	71021	49589	19240.53	1.92E-05	0.080401
registerDevice	189105	167041	64811.91	6.48E-05	0.270831
transferDeviceOwnership	32085	10145	3936.26	3.94E-06	0.016449
notifyReceivedDevice	32085	3124	1212.112	1.21E-06	0.005065

throughout their supply chains. Moreover, the one-time execution nature of the proposed blockchain functions highlights the cost-effectiveness and efficiency of using blockchain technology to establish trust and accountability in supply chain operations. The `PCBVerification` smart contract is carefully designed to minimize gas consumption by leveraging structures and mappings, with mappings enabling constant-time data access via hash tables, and access-control modifiers restricting function execution to authorized nodes, thereby reducing unnecessary operations and saving gas [12]. Events are used to log actions without persisting data on-chain, further lowering gas costs compared to state storage, while expensive on-chain storage of large datasets and computation-intensive tasks, such as machine learning inference, are deliberately avoided by performing these operations off-chain at the verifier's facility and recording only the inference outcomes on the blockchain. Although constructs such as loops and `keccak256` hashing can increase gas costs [21], their usage is kept to a minimum.

To additionally ensure privacy, integrity, and post-quantum security, a fully encrypted blockchain implementation based on Zama's FHEVM framework [37] is presented in detail in Appendix G, through the `EncryptedPCBVerifier` contract, which maintains all critical state variables—including voting results, device prices, and counterfeit indicators—in encrypted form throughout input, execution, and storage. Using fully homomorphic encryption, arithmetic operations, logical comparisons, and conditional branching are performed directly on ciphertexts without intermediate decryption, with security grounded in the hardness of the Learning With Errors (LWE) problem, thereby providing resilience against quantum adversaries capable of breaking classical cryptosystems such as RSA or ECC. Privacy is preserved even in a permissionless blockchain setting, as validators cannot access plaintext data and selective decryption is granted only to authorized entities via the `TFHE.allow` mechanism. The source codes for both the `PCBVerifier` and the post-quantum secure `EncryptedPCBVerifier` smart contracts are added in a GitHub repository[2].

When a new PCB class enters the supply chain, the Trusted Authority (TA) employs an incremental transfer-learning strategy instead of retraining the CNN

[2] `https://github.com/shuvodipmaitra/PCB_Verification_Blockchain`

from scratch, wherein the pre-trained backbone is frozen and stored in memory, while only the final classification layers are fine-tuned on images of the new class, enabling continuous deployment through inference on the latest model checkpoint. A micro-focus XCT system scans a 50×50 mm PCB in approximately 1.5–2 hours; for high-volume industrial operation, the framework can be extended to fast 2D X-ray radiographs, sparse-view CT, and parallelized gantry units, ensuring compatibility with commercial inline X-ray inspection systems. Incremental updates are rapid and low-cost, with validated model versions stored off-chain and automatically fetched by TA nodes, while even initial model training remains inexpensive, requiring only 5–10 hours of GPU training on approximately 9000 XCT images. The typical GPU operating costs of 0.056–0.11 USD/hour result in a total training cost not exceeding 1 USD, assuming existing infrastructure already in place. A tabular comparison with existing PCB supply-chain methods is provided in Table 5 of Appendix B. The associated smart contract is optimized to minimize storage writes and stores only lightweight metadata, including class identifiers, authenticity flags, and Grad-CAM hashes, thereby maintaining negligible operational costs while enforcing role-based access control. Dispute resolution is handled via an on-chain voting protocol among verified stakeholders, coordinated by a master TA node that enforces global standardization of XCT acquisition parameters and model specifications, yielding an imaging-based verification framework that is both technically and economically scalable.

7 Conclusion

In this paper, we present a blockchain-based framework for PCB counterfeit detection in which a PCB verifier employs XCT scanning and a machine learning based classification pipeline to authenticate PCB devices. Acting as a trusted authority (TA) node, the PCB verifier interacts with multiple PCB OEMs and vendors as peer nodes within the blockchain network and deploys a smart contract that records verification information for every tested PCB in a transparent and immutable manner. The combination of accurate machine learning–based classification and low gas consumption demonstrates the practicality and scalability of the proposed approach, indicating its suitability for large-scale deployment to mitigate PCB counterfeiting, a problem that has led to substantial losses across the PCB supply chain.

8 Acknowledgment

This work is partially supported by the Centre on Hardware Security Entrepreneurship Research and Development (CHERD) project (HSE), Information Security Education and Awareness Project (YAP), funded by Ministry of Electronics and Information Technology (MeitY), India; and the Development of Secured Hardware and Automotive Systems (DSY) project, funded by IHUB NTIHAC Foundation, Department of Science and Technology (DST), India.

References

1. Akcay, S., Breckon, T.P.: An evaluation of region based object detection strategies within x-ray baggage security imagery. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 1337–1341 (2017). https://doi.org/10.1109/ICIP.2017.8296499
2. Akçay, S., Kundegorski, M.E., Devereux, M., Breckon, T.P.: Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 1057–1061 (2016). https://doi.org/10.1109/ICIP.2016.7532519
3. Botero, U.J., Ganji, F., Asadizanjani, N., Woodard, D.L., Forte, D.: Semi-supervised automated layer identification of x-ray tomography imaged pcbs. In: 2020 IEEE Physical Assurance and Inspection of Electronics (PAINE). pp. 1–6 (2020). https://doi.org/10.1109/PAINE49178.2020.9337738
4. Buterin, V.: Ethereum: A next-generation smart contract and decentralized application platform (2014), https://github.com/ethereum/wiki/wiki/White-Paper, accessed: 2016-08-22
5. Castillo, E., Meyer-Baese, U., García, A., Parrilla, L., Lloris, A.: Ipp@hdl: efficient intellectual property protection scheme for ip cores. IEEE Trans. Very Large Scale Integr. Syst. **15**(5), 578–591 (may 2007). https://doi.org/10.1109/TVLSI.2007.896914, https://doi.org/10.1109/TVLSI.2007.896914
6. Chaudhary, C.K., Chatterjee, U., Mukhopadhayay, D.: Auto-pufchain: An automated interaction tool for pufs and blockchain in electronic supply chain. In: 2021 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). pp. 1–4 (2021). https://doi.org/10.1109/AsianHOST53231.2021.9699720
7. Chollet, F.: Xception: Deep learning with depthwise separable convolutions (2017), https://arxiv.org/abs/1610.02357
8. Franceschi, L., et al.: Robustness of classifiers to uniform lp and gaussian noise. In: Proceedings of the 35th International Conference on Machine Learning. pp. 1597–1606. PMLR (2018)
9. Harrison, J., Asadizanjani, N., Tehranipoor, M.: On malicious implants in pcbs throughout the supply chain. Integration **79**, 12–22 (2021). https://doi.org/https://doi.org/10.1016/j.vlsi.2021.03.002, https://www.sciencedirect.com/science/article/pii/S0167926021000304
10. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2018), https://arxiv.org/abs/1608.06993
11. Innovatiana: The importance of noise in machine learning. https://www.innovatiana.com/en/post/add-noise-in-ai (2024), accessed: 2025-09-25
12. Khanzadeh, S., Samreen, N., Alalfi, M.H.: Optimizing gas consumption in ethereum smart contracts: Best practices and techniques. In: 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C). pp. 300–309 (2023). https://doi.org/10.1109/QRS-C60940.2023.00056
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Lee, J.H., Pilkington, M.: How the blockchain revolution will reshape the consumer electronics industry [future directions]. IEEE Consumer Electronics Magazine **6**(3), 19–23 (2017). https://doi.org/10.1109/MCE.2017.2684916
15. Lee, J., Lim, D., Gassend, B., Suh, G., van Dijk, M., Devadas, S.: A technique to build a secret key in integrated circuits for identification and authentication

applications. In: 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525). pp. 176–179 (2004). https://doi.org/10.1109/VLSIC.2004.1346548
16. Liang, K.J., Sigman, J.B., Spell, G.P., Strellis, D., Chang, W., Liu, F., Mehta, T., Carin, L.: Toward automatic threat recognition for airport x-ray baggage screening with deep convolutional object detection (2019), https://arxiv.org/abs/1912.06329
17. Lin, C.J., Jhang, J.Y.: Bearing fault diagnosis using a grad-cam-based convolutional neuro-fuzzy network. Mathematics **9**(13) (2021). https://doi.org/10.3390/math9131502, https://www.mdpi.com/2227-7390/9/13/1502
18. Ma, B., Jia, T., Su, M., Jia, X., Chen, D., Zhang, Y.: Automated segmentation of prohibited items in x-ray baggage images using dense de-overlap attention snake. IEEE Transactions on Multimedia **25**, 4374–4386 (2023). https://doi.org/10.1109/TMM.2022.3174339
19. Mehta, D., Lu, H., Paradis, O.P., M. S., M.A., Rahman, M.T., Iskander, Y., Chawla, P., Woodard, D.L., Tehranipoor, M., Asadizanjani, N.: The big hack explained: Detection and prevention of pcb supply chain implants. J. Emerg. Technol. Comput. Syst. **16**(4) (aug 2020). https://doi.org/10.1145/3401980, https://doi.org/10.1145/3401980
20. Miao, C., Xie, L., Wan, F., Su, c., Liu, H., Jiao, j., Ye, Q.: Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. In: CVPR (2019)
21. Naik, R.P., Courtois, N.T.: Optimising the SHA256 Hashing Algorithm for Faster and More Efficient Bitcoin Mining. PhD Thesis, University College London (September 2013), http://www.nicolascourtois.com/bitcoin/Optimising%20the%20SHA256%20Hashing%20Algorithm%20for%20Faster%20and%20More%20Efficient%20Bitcoin%20Mining_Rahul_Naik.pdf
22. Rajendran, J., Sam, M., Sinanoglu, O., Karri, R.: Security analysis of integrated circuit camouflaging. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. p. 709–720. CCS '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2508859.2516656, https://doi.org/10.1145/2508859.2516656
23. Rajendran, J., Sinanoglu, O., Karri, R.: Is split manufacturing secure? In: 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1259–1264 (2013). https://doi.org/10.7873/DATE.2013.261
24. Rogers, T.W., Jaccard, N., Griffin, L.D.: A deep learning framework for the automated inspection of complex dual-energy x-ray cargo imagery. In: Ashok, A., Franco, E.D., Gehm, M.E., Neifeld, M.A. (eds.) Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 10187, p. 101870L (May 2017). https://doi.org/10.1117/12.2262662
25. Sarkar, T.S., Maitra, S., Chakraborty, A., Bhattacharya, S., Mukhopadhyay, D.: Trex-f: Trustability of electronics using x-ray based fingerprinting. In: 2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD). pp. 1–9 (2025). https://doi.org/10.1109/ICCAD66269.2025.11240899
26. Sarkar, T.S., Maitra, S., Chakraborty, A., Saha, A., Chowdhury, J., Mukhopadhyay, D.: X-factor: Deep learning-based pcb counterfeit detection using x-ray ct techniques for hardware assurance. In: Proceedings of the 21st ACM International Conference on Computing Frontiers: Workshops and Special Sessions. p. 25–34. CF '24 Companion, Association for Computing Machinery, New York, NY,

USA (2024). https://doi.org/10.1145/3637543.3654657, https://doi.org/10.1145/3637543.3654657

27. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017). https://doi.org/10.1109/ICCV.2017.74
28. Sen, J.: Fgsm and patch attacks and their impact (2023), https://arxiv.org/pdf/2307.02055.pdf
29. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE Design Automation Conference. pp. 9–14 (2007)
30. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594
31. Tsiligkaridis, T., Tsiligkaridis, A.: Diverse gaussian noise consistency regularization for robustness and uncertainty calibration. arXiv preprint arXiv:2104.01231 (2021)
32. Vosatka, J., Stern, A., Hossain, M., Rahman, F., Allen, J., Allen, M., Farahmandi, F., Tehranipoor, M.: Confidence modeling and tracking of recycled integrated circuits, enabled by blockchain. In: 2020 IEEE Research and Applications of Photonics in Defense Conference (RAPID). pp. 1–3 (2020). https://doi.org/10.1109/RAPID49481.2020.9195666
33. Waghela, H., Sen, J., Rakshit, S.: Robust image classification: Defensive strategies against fgsm and pgd adversarial attacks. In: Proceedings of the 2024 Asian Conference on Intelligent Technologies (ACOIT) (2024), https://arxiv.org/abs/2408.13274
34. Xu, H., Wang, Y., Liu, P., Chen, P.Y., Zhang, W., Zhang, T.: Adversarial attacks and defenses in images, graphs and text: A review. arXiv preprint arXiv:1909.08072 (2019), https://arxiv.org/pdf/1909.08072.pdf
35. Xu, X., Rahman, F., Shakya, B., Vassilev, A., Forte, D., Tehranipoor, M.: Electronics supply chain integrity enabled by blockchain. ACM Trans. Des. Autom. Electron. Syst. **24**(3) (may 2019). https://doi.org/10.1145/3315571, https://doi.org/10.1145/3315571
36. Yang, J., Zhao, Z., Zhang, H., Shi, Y.: Data augmentation for x-ray prohibited item images using generative adversarial networks. IEEE Access **7**, 28894–28902 (2019). https://doi.org/10.1109/ACCESS.2019.2902121
37. Zama: FHEVM: A full-stack framework for integrating Fully Homomorphic Encryption (FHE) with blockchain applications. https://github.com/zama-ai/fhevm (2025), accessed: 2025-09-26
38. Zhang, L., Jiang, L., Ji, R., Fan, H.: Pidray: A large-scale x-ray benchmark for real-world prohibited item detection (2022), https://arxiv.org/abs/2211.10763
39. Zhao, Z., Zhang, H., Yang, J.: A gan-based image generation method for x-ray security prohibited items. In: Pattern Recognition and Computer Vision: First Chinese Conference, PRCV 2018, Guangzhou, China, November 23-26, 2018, Proceedings, Part I. p. 420–430. Springer-Verlag, Berlin, Heidelberg (2018). https://doi.org/10.1007/978-3-030-03398-9_36, https://doi.org/10.1007/978-3-030-03398-9_36

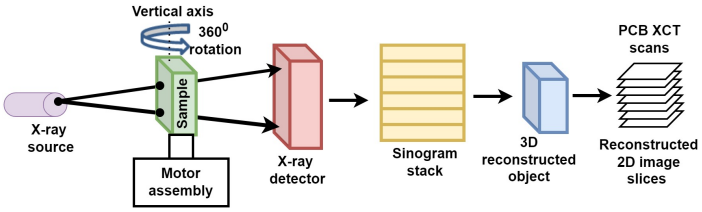


Fig. 11: XCT Image Acquisition

A Image acquisition set-up

The XRM set-up (as shown in Fig. 11) has an X-ray source capable of delivering about 160 kV and 10 W power. The detector assembly is composed of a scintillator and four different objective lenses: 0.4×, 4×, 20× and 40×. By controlling the source-to-sample and sample-to-detector distance, we can adjust the voxel size in the projection images. The source exposure time must be adjusted to yield the desired intensity count of the projection image. The XRM acquires 2D projection images of the PCB samples at angles from 0 to 360^0. We get a stack of CT sinograms where each layer represents a particular angle's X-ray projection. Adjusting the angular steps during the rotation process, we acquire 1601 such X-ray projection images. Using these projection images and applying a filter-back propagation algorithm, a 3D image of the PCB sample is constructed. The number of projection images is optimally chosen based on the quality of reconstructed data that we want, and the minimum number of images that the filter-back projection technique requires.

B Comparison of X-CHAIN framework with other methods used in Electronic Supply Chain

Herein, we present a Tabular comparison (Table 5) of our X-CHAIN framework with other techniques adopted for electronics supply chain assurance.

C Advantages of Full view XCT

We would also like to emphasize the fact that full-view XCT (that we have used in this paper) offers significantly superior imaging quality compared to 2D radiographs and sparse-view XCT, which is critical for reliable PCB authentication. 2D X-ray radiographs collapse all internal layers (of the PCB sample) into a single projection and suffer from high noise and depth-superposition artefacts. Sparse-view XCT introduces streaking, blurring, and limited-angle reconstruction errors. In comparison, full-view XCT provides high signal-to-noise ratio (SNR), artefact-free volumetric data with consistent voxel resolution. This enables clear separation of multilayer routing, vias, and subtle internal defects,

Table 5: X-CHAIN Comparison Table

Method	Capabilities	Limitations
Automated Optical Inspection (AOI), Automated X-ray Inspection (AXI)	Surface-level inspection is good for solder joints and assembly defects.	Cannot observe internal copper traces, buried vias, or multilayer structures
Electrical Tests: In-Circuit Test (ICT) Functional Circuit Test (FCT)	Validates electrical connectivity and functionality via test points.	Requires fixtures; cannot detect internal structural tampering; invasive and time-consuming for diverse PCB models.
Destructive Physical Analysis	Full visibility of internal layers and copper patterns.	Destroys the sample; unsuitable for supply-chain deployments; costly and slow.
Certificate-based Digital Identification (e.g., barcode, RFID)	Tracks device identity and ownership; low operational cost.	Cannot detect physical counterfeits; relies entirely on documentation, which can be forged or bypassed.
X-CHAIN Framework (Ours)	Non-destructive internal structure verification; DL-based authenticity classification; Grad-CAM anomaly localization; immutable blockchain backed traceability	Requires XCT equipment; moderate scan time for high-volume deployment.

allowing our deep-learning and Grad-CAM-based anomaly detection pipeline to accurately localize and interpret suspicious regions. Although full XCT is slower, these substantial gains in image quality, noise reduction, and artefact suppression make it the most reliable modality for imaging-based high-assurance, non-destructive PCB verification.

D Grad-CAM heatmap validation; and operational impacts of false positive and false negative

Grad-CAM maps are employed not as a standalone decision mechanism but as an interpretability and forensic-support tool to explain why the one-class SVM flags a PCB as suspicious, where slice-level heatmaps highlight feature regions contributing strongly to the anomaly score, and the stacked 3D heatmap enables tracking of these patterns across the PCB volume. The highlighted regions are validated by cross-referencing with known structural anomalies such as misaligned internal layers, irregular via geometry, unexpected copper discontinuities, void patterns, and defects. Since the dataset is generated from fully reconstructed XCT volumes, the Grad-CAM regions can be directly compared with the ground-truth physical structure visible in the slices. During off-chain model updates, TA nodes further evaluate the updated model on validation PCBs with operator-confirmed anomalies to ensure that the Grad-CAM patterns correspond to meaningful physical indicators rather than noise or artefacts.

In high-assurance electronics supply chains, false positives and false negatives have asymmetric consequences. False positives may delay shipment or require secondary inspection but incur limited operational overhead, which is mitigated through high-quality full-view XCT imaging, the two-level DL classification pipeline, and Grad-CAM-based 3D visualization that enables rapid human review. False negatives pose significantly higher economic and operational risks by allowing counterfeit PCBs to propagate into critical systems such as industrial controllers, medical devices, defense electronics, and automotive ECUs, potentially causing failures, safety hazards, or supply-chain compromise. Thus, the use of full XCT volumes instead of 2D or sparse-view scans is justified, as full XCT minimizes false negatives by providing high-SNR, artifact-free visibility of internal layers, vias, and hidden modifications in the PCB sample.

E Adversarial training and Noise addition for Second-level classification

Tables 6 and 7 report the performance of the proposed DenseNet-121 feature extraction and one-class SVM pipeline under FGSM and PGD adversarial augmentation, respectively. For FGSM augmentation with perturbation strengths $\epsilon \in [0.1, 0.5]$, Precision remains consistently perfect (1.0) across all settings, indicating the absence of false positives, while Recall and F1-score vary with both ϵ and the SVM parameter ν. Higher ν values (0.95 and 0.90) yield perfect Recall and F1, whereas reducing ν makes the SVM decision boundary more restrictive, lowering Recall to approximately 0.89–0.97. Increasing ϵ from 0.1 to 0.5 improves Recall and F1, particularly at lower ν, demonstrating that stronger FGSM-based adversarial augmentation enhances generalization. Overall, $\epsilon = 0.3$–0.4 with $\nu \geq 0.70$ offers the best trade-off.

Similarly, PGD-based augmentation in Table 7 achieves a precision of 1.0 for all configurations, while Recall and F1 depend on ϵ, step size α, and ν. Higher ν values (0.95 and 0.90) results in Recall and F1-scores above 0.98, whereas decreasing ν substantially reduces Recall (e.g., from 0.9742 at $\nu = 0.90$ to around 0.68 at $\nu = 0.60$). Moderate increases in α (e.g., $\alpha = 0.05$) improve Recall at intermediate ν, indicating better generalization from stronger PGD perturbations, while increasing ϵ beyond 0.2 yields no consistent benefit. Overall, PGD augmentation with $\epsilon = 0.1$–0.2, $\alpha = 0.05$, and $\nu \geq 0.90$ provides the most favorable balance between Recall and F1.

Tables 8a and 8b analyze the effect of stochastic noise-based data augmentation, including Gaussian noise and salt-and-pepper noise, on one-class SVM performance using DenseNet-121 features. For Gaussian noise with variances $\sigma^2 = 10, 15, 20, 30$ in Table 8a, Precision remains 1.0 across all ν, confirming zero false positives, while Recall and F1-score decrease slightly as ν is reduced due to stricter decision boundaries (e.g., Recall = 0.9474 at $\sigma^2 = 10$, $\nu = 0.60$). Notably, higher noise variances marginally improve robustness at lower ν (e.g., Recall = 0.9653 and F1 = 0.9823 at $\sigma^2 = 30$, $\nu = 0.60$), indicating enhanced generalization from richer feature distributions.

Table 6: FGSM adversarial training results. Precision is 1.0 for all configurations.

ϵ	Metric	ν=0.95	ν=0.90	ν=0.80	ν=0.70	ν=0.60
0.1	Recall	1.000	1.000	0.999	0.973	0.894
	F1	1.000	1.000	0.995	0.986	0.944
0.2	Recall	1.000	1.000	0.998	0.981	0.927
	F1	1.000	1.000	0.999	0.997	0.962
0.3	Recall	1.000	1.000	0.999	0.987	0.947
	F1	1.000	1.000	1.000	0.994	0.973
0.4	Recall	1.000	1.000	1.000	0.995	0.963
	F1	1.000	1.000	1.000	0.995	0.981
0.5	Recall	1.000	1.000	1.000	0.999	0.976
	F1	1.000	1.000	1.000	1.000	0.998

Table 7: PGD adversarial training results for varying perturbation budgets $\epsilon \in \{0.1, 0.2, 0.3, 0.4\}$ and step sizes $\alpha \in \{0.01, 0.02, 0.05\}$. Precision is 1.0 for all configurations.

ϵ	α	Metric	ν=0.95	ν=0.90	ν=0.80	ν=0.70	ν=0.60
0.1	0.01	Recall	1.000	0.974	0.859	0.768	0.681
		F1	1.000	0.986	0.924	0.869	0.810
	0.02	Recall	1.000	0.963	0.842	0.770	0.686
		F1	1.000	0.981	0.914	0.868	0.814
	0.05	Recall	1.000	0.985	0.886	0.808	0.730
		F1	1.000	0.993	0.940	0.894	0.844
0.2	0.01	Recall	1.000	0.974	0.858	0.770	0.679
		F1	1.000	0.987	0.924	0.870	0.809
	0.02	Recall	1.000	0.959	0.837	0.775	0.695
		F1	1.000	0.979	0.911	0.873	0.820
	0.05	Recall	1.000	0.967	0.857	0.787	0.727
		F1	1.000	0.983	0.923	0.881	0.842
0.3	0.01	Recall	1.000	0.973	0.861	0.770	0.683
		F1	1.000	0.986	0.925	0.870	0.811
	0.02	Recall	1.000	0.956	0.835	0.763	0.691
		F1	1.000	0.978	0.910	0.866	0.817
	0.05	Recall	1.000	0.963	0.842	0.780	0.705
		F1	1.000	0.981	0.914	0.876	0.827
0.4	0.01	Recall	1.000	0.973	0.856	0.769	0.678
		F1	1.000	0.986	0.923	0.869	0.808
	0.02	Recall	1.000	0.956	0.834	0.761	0.692
		F1	1.000	0.978	0.910	0.864	0.818
	0.05	Recall	1.000	0.955	0.837	0.779	0.701
		F1	1.000	0.977	0.911	0.876	0.825

Table 8: Performance comparison of noise-based data augmentation methods

(a) Gaussian noise addition

Gaussian parameter (σ^2)	SVM parameter (ν)	Precision	Recall	F1
10	0.95	1.0	1.0	1.0
	0.90	1.0	1.0	1.0
	0.80	1.0	1.0	1.0
	0.70	1.0	0.9871	0.9935
	0.60	1.0	0.9474	0.9740
15	0.95	1.0	1.0	1.0
	0.90	1.0	1.0	1.0
	0.80	1.0	0.9940	0.9970
	0.70	1.0	0.9891	0.9945
	0.60	1.0	0.9544	0.9766
20	0.95	1.0	1.0	1.0
	0.90	1.0	1.0	1.0
	0.80	1.0	0.9940	0.9970
	0.70	1.0	0.9881	0.9940
	0.60	1.0	0.9613	0.9803
30	0.95	1.0	1.0	1.0
	0.90	1.0	1.0	1.0
	0.80	1.0	0.9980	0.9990
	0.70	1.0	0.9901	0.9950
	0.60	1.0	0.9653	0.9823

(b) Salt and pepper noise addition

Salt prob.	Pepper prob.	SVM param (ν)	Prec.	Recall	F1
0.02	0.02	0.95	1.0	1.0	1.0
		0.90	1.0	1.0	1.0
		0.80	1.0	0.9891	0.9945
		0.70	1.0	0.9692	0.9844
		0.60	1.0	0.9028	0.9489
0.02	0.05	0.95	1.0	1.0	1.0
		0.90	1.0	1.0	1.0
		0.80	1.0	0.9980	0.9990
		0.70	1.0	0.9742	0.9869
		0.60	1.0	0.9256	0.9614
0.05	0.02	0.95	1.0	1.0	1.0
		0.90	1.0	1.0	1.0
		0.80	1.0	0.9960	0.9980
		0.70	1.0	0.9643	0.9818
		0.60	1.0	0.9107	0.9533
0.05	0.05	0.95	1.0	1.0	1.0
		0.90	1.0	1.0	1.0
		0.80	1.0	1.0	1.0
		0.70	1.0	0.9802	0.9900
		0.60	1.0	0.9306	0.9640

In Table 8b, salt-and-pepper noise augmentation also preserves perfect Precision across all configurations, with Recall gradually declining as ν decreases. Low noise probabilities (e.g., salt = 0.02, pepper = 0.02) maintain near-perfect performance, while higher probabilities (e.g., 0.05) introduce additional challenge but still achieve strong F1-scores (approximately 0.95 at $\nu = 0.60$). These results demonstrate that both Gaussian and impulsive noise augmentation yield stable and highly precise representations, confirming the robustness and generalization capability of the DenseNet-121 and one-class SVM framework under diverse perturbation models.

F Ethereum Smart Contract Details

In the Ethereum-based private blockchain, each node is uniquely identified by a 160-bit address and is assigned a permission level to regulate access and functionality. Trusted Authority (TA) nodes are assigned permission level 1, PCB OEM nodes are assigned level 2, and PCB vendor nodes are assigned level 0. The smart contract constructor initializes the system by registering the initial TA nodes with their timestamps, identifying the most senior TA node as the master TA, initializing state variables, and transferring contract ownership to the master TA. Seniority is determined by iterating over all initial TA nodes and comparing their registration timestamps. The master TA can register new TA nodes while maintaining their active status, timestamps, and total count.

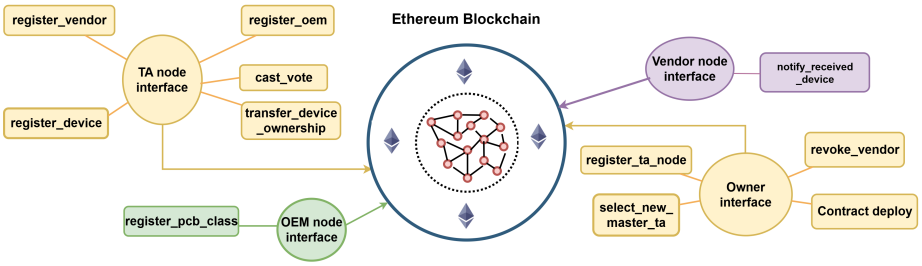


Fig. 12: Web interfacing with blockchain

A voting-based mechanism enables the election of a new master TA node, in which only the active TA nodes can vote within a predefined period. Votes are recorded, and the new master TA is selected based on the highest vote count, with seniority as a tie-breaker. Upon selection, ownership is transferred and an event is emitted; this process can only be triggered by the current contract owner. The TA nodes are also responsible for registering new PCB OEMs and vendors (in their geographical area), assigning them appropriate permission levels (2 and 0, respectively), and emitting corresponding registration events.

Registered OEMs can add new PCB classes by assigning a unique class ID, class name, and OEM address, incrementing the total class count, and emitting an event. Vendors notify the blockchain upon receiving new PCB devices by emitting events containing device IDs and vendor addresses, enabling nearby TA nodes to monitor supply-chain activity. TA nodes subsequently register these devices on-chain by creating a `DeviceInfo` structure that includes the device ID, vendor and verifier addresses, class name, counterfeit value, device figure, and device price tokens. The device ID is uniquely generated by hashing the 3D heatmap image obtained from the Grad-CAM process. Device ownership transfer between sellers and customers is managed by TA nodes, subject to the conditions that the device is not counterfeit and the buyer has sufficient balance. Additionally, the master TA node can revoke a vendor's registration if the vendor possesses at least five counterfeit devices.

Interaction with the blockchain is facilitated through a Python web interface built using the `web3` library, enabling TA nodes, OEMs, vendors, and the owner to deploy and interact with the smart contract. The interface connects to a local Ethereum node via `HTTPProvider`, deploys the contract using the extracted ABI and bytecode, retrieves the contract address from the transaction receipt, and executes subsequent function calls by sending transactions and awaiting confirmations, ensuring reliable and transparent blockchain operations.

G Post-quantum secure blockchain implementation

We present a proof-of-concept post-quantum secure and privacy-preserving blockchain framework, termed `EncryptedPCBVerifier`, implemented in `Solidity`

and deployed on Zama's Fully Homomorphic Encryption Virtual Machine (FHEVM). This design preserves the complete functionality of the previously described Ethereum-based PCB verification system while ensuring that all sensitive on-chain data—including PCB class identifiers, pricing information, counterfeit indicators, permissions, balances, votes, and device identifiers—remain encrypted at all times. Built on the `TFHE` (Torus Fully Homomorphic Encryption) scheme, the FHEVM enables arithmetic, comparison, and conditional operations to be performed directly on ciphertexts without ever revealing plaintext values on-chain. The security of TFHE relies on the hardness of the Learning With Errors (LWE) problem, providing strong post-quantum guarantees beyond classical cryptographic schemes such as RSA or ECC.

To ensure input integrity, the system employs zero-knowledge proofs (e.g., `timestampProof`, `classNameProof`) that validate encrypted inputs without disclosure, while fine-grained access control is enforced via `TFHE.allow`, preventing miners and validators from decrypting sensitive data. The contract integrates `TFHE.sol`, which introduces encrypted data types such as `euint8`, `euint32`, `euint64`, and `ebool` supporting homomorphic arithmetic, comparisons, conditional selection, and permission management, alongside `Ownable2Step.sol` from OpenZeppelin to ensure secure two-step administrative ownership transfer.

During initialization, the contract accepts encrypted registration timestamps for initial Trusted Authority (TA) nodes, verifies them via zero-knowledge proofs, converts them to encrypted types, and stores encrypted vote counts and timestamps while selectively granting decryption rights to authorized entities. The most senior initial TA node is determined using encrypted comparisons and conditional selection, establishing the first master TA without revealing any registration timestamps. The master TA can register new TA nodes, increment encrypted counters, and emit events without leaking internal state. Encrypted voting allows active TA nodes to cast votes within a defined voting period, and a new master TA is selected by comparing encrypted vote counts and encrypted registration timestamps in case of ties. TA nodes can privately register OEMs, vendors, and PCB classes using encrypted permissions, class identifiers, and class names, while vendors notify device reception using encrypted device IDs. TA nodes register devices using encrypted class, price, and counterfeit indicators, and manage encrypted ownership and balances during ownership transfer. The master TA node can also revoke vendors by counting encrypted counterfeit occurrences exceeding a threshold value of five.

Secure web interfacing in the FHEVM architecture combines a frontend, backend API, and FHEVM client, where users encrypt inputs locally, submit encrypted transactions with proofs, and decrypt outputs locally if authorized. All interactions rely on a global FHE key securely distributed to clients and never embedded in the smart contract. This ensures that the complete PCB supply-chain lifecycle remains verifiable, confidential, and post-quantum secure throughout the on-chain execution.