ORIGINAL ARTICLE

Journal Section

A Hierarchical Learning and Security Framework for Anomaly Detection in Autonomous Underwater Vehicles

Gurvir Singh[1*] | Dheeraj Panwar[2*] | Devesh Panwar[2*] | Shilpi Mittal[3*]

[1]University Institue of Computing, Chandigarh University, Punjab, India

[2]University Institue of Computing, Chandigarh University, Punjab, India

[2]University Institue of Computing, Chandigarh University, Punjab, India

[3]University Institue of Computing, Chandigarh University, Punjab, India

Correspondence
Shilpi Mittal, University Institute of Computing, Chandigarh University, Punjab, India
Email: g.shilpi84@gmail.com

Funding information

The increasing automation of Autonomous Underwater Vehicles (AUVs) requires the development of intrusion detection systems that can operate under system and communication constraints. Traditional centralized approaches often fail under such circumstances because they are not designed to operate under bandwidth limitations, data privacy concerns, and vulnerability to cyber-attacks. The purpose of this paper is to suggest a Hierarchical Learning and Security Framework that combines Federated Learning (FL), Blockchain-based aggregation, and Knowledge Distillation (KD) to tackle these problems. In this proposed approach, AUV clients perform local anomaly detection using lightweight models that can work under any given system constraints, while collaboratively training a global model through federated learning. The implementation of blockchain ensures that the model updates are tamper-free and verifiable. Knowledge Distillation transfers knowledge from a large-scale high-capacity trained model to the low-scale model, improving its accuracy so that it can predict and detect attacks better even while under low resource

Abbreviations: AUV, Autonomous Underwater Vehicle; FL, Federated Learning; KD, Knowledge Distillation; IoT, Internet of Things; SFI Smart Ocean Dataset

[*] Both 2*nd authors contributed equally.

constraints and using the same lightweight model. Experimental evaluation was conducted on a publicly available AUV dataset (SFI Smart Ocean Dataset) with multiple sensor nodes, using simulated Federated Learning, Blockchain, and KD environments. A significant improvement was observed in the same lightweight model's detection of threats (from a baseline of 45.8 percent to 62.9 percent), while reducing communication overhead and preserving client data privacy. The integration of Knowledge Distillation further improves the performance of low-resource clients without compromising the efficiency of the lightweight models. This study provides a scalable and secure methodology for anomaly detection in AUVs, with potential implementation in broader IoT-driven autonomous systems such as electric vehicles and underwater communication networks.

KEYWORDS
Autonomous Underwater Vehicles, Anomaly Detection, Federated Learning, Blockchain, Knowledge Distillation, IoT Security

1 | INTRODUCTION

Autonomous Underwater Vehicles (AUVs) are rapidly becoming very important assets in domains ranging from oceanographic research and environmental monitoring to ocean infrastructure inspection and defense operations. As these machines explore into increasingly complex and remote environments , their ability to operate without constant human oversight becomes more and more crucial. A critical component of this autonomy is their capacity for real time on-device intrusion and anomaly detection , which enables an AUV to identify system failures , hazards and far above , any cyber-attack attempts on it thus ensuring mission safety an data integrity [1]. However, deploying effective machine learning model for AUVs presents a significant set of challenges. The hardware on these platforms is very resource limited and does not possess the heavy computational power and energy supply, stopping the possibility of using a high accuracy deep learning model .

Furthermore, even if the model is collaboratively trained, it is still limited by the inherent simplicity of the lightweight architecture of the model that is used to run the AUVs. This performance gap between low scale models that can run on small scale hardware and high end models that can detect and predict far better but require much stronger and larger hardware is a major hurdle [2]. Knowledge Distillation (KD) offers a major solution, where a large, complex "teacher" model often a large scale deep learning model trained on the same type of data on a large scale transfers its learned knowledge to a smaller "student" model, significantly boosting the student's performance without increasing its size or computational cost [3]. By integrating KD into our workflow , we can improve the accuracy of the smaller model without requiring any more computational power. As shown in Figure 1 , it clearly defines the method-

olgy of the workflow used and how the combined structure functions as a whole. This paper proposes a framework that combines these 3 vital things – Federated Learning ,Blockchain and Knowledge Distillation to create a secure and high performance system for AUVs of all types while keeping constraints of the machines computational power in view. We detail the system's architecture, the mathematical foundations of its learning protocols, and provide a comprehensive evaluation that validates its effectiveness.

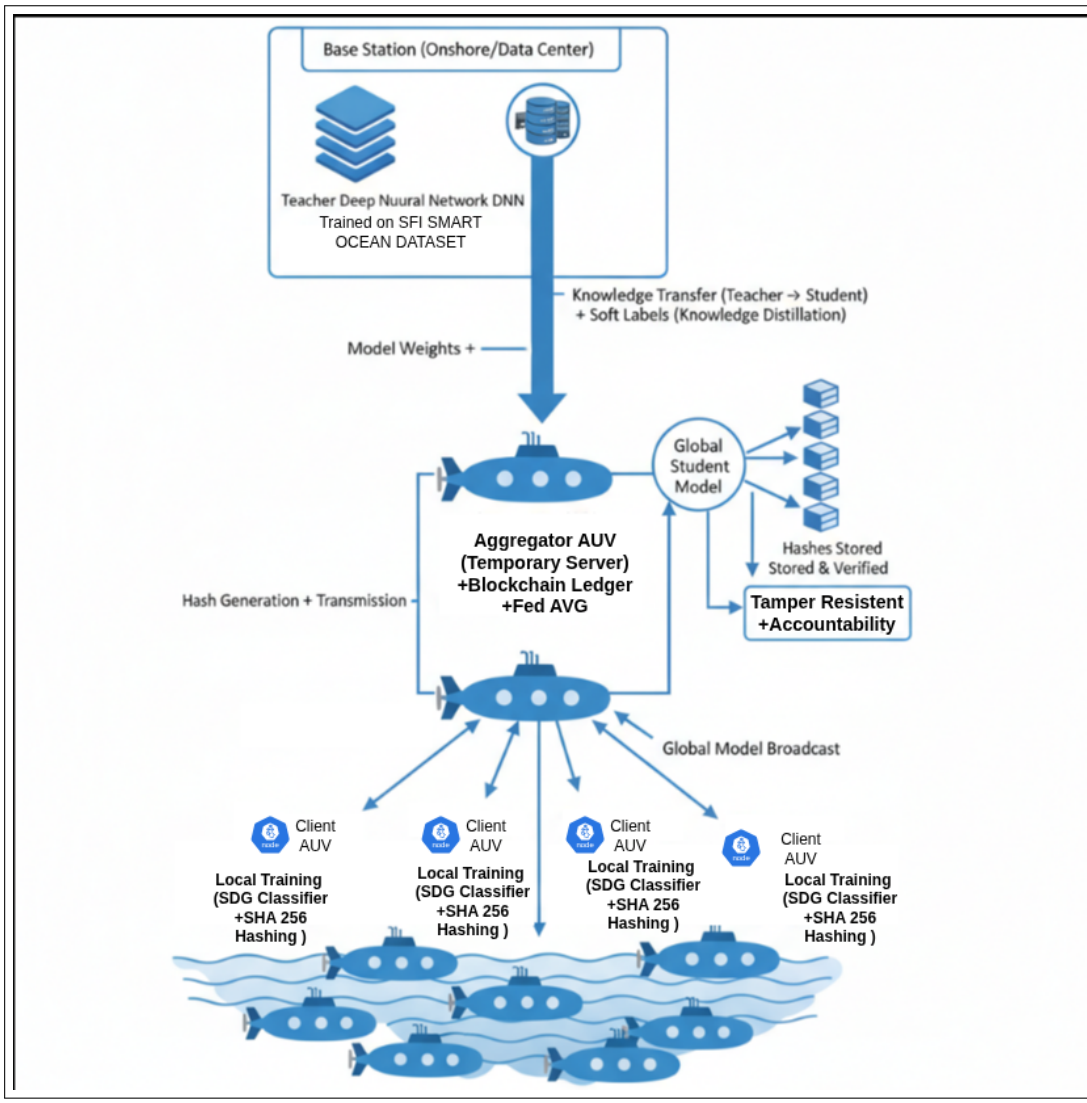

FIGURE 1 Blockchain-secured federated learning workflow for an AUV swarm with teacher–student knowledge distillation.

1.1 | Motivation

The Primary Motivation and need for this work is the need to address the critical gap in current AUV systems and capabilities and the need for a high accuracy on-device anomaly detection system that can operate under strict computational constraints which not only guaranteeing good accuracy but also privacy of the learning process and results itself. Conventional approaches often force a trade-off between model performance and deployability, and if not, they overlook the security risks of such anomaly detection systems. Our Research into this topic is driven by this need of a holistic framework that solves these conflicting requirements , enabling the deployment of truly intelligent and resilient AUV swarms.

1.2 | Contribution

The main contributions of this work are as follows:

- We propose a novel, three-stage hierarchical framework that integrates Federated Learning (FL), Blockchain, and Knowledge Distillation for secure anomaly detection systems in a resource constrained environment which possess both security and accuracy.
- We introduce a blockchain-based integrity verification layer into the federated learning process, where cryptographic hashes of model updates are stored on an immutable ledger to protect against tampering and ensure the provenance of all contributions.
- We demonstrate the profound impact of Knowledge Distillation in this domain, showing its ability to transfer knowledge from a larger deep learning model teacher to a student model which is usually a low accuracy model , boosting its accuracy drastically on the same low scale model.
- We provide a comprehensive performance evaluation based on simulations using a large, balanced real-world dataset (SFI Smart Ocean) establishing the claim that findings can be actually used in the industry to make AUVs more functional .

2 | RELATED WORK

Anomaly Detection in AUVs: Raanan et al. [4] researched in AUV self-perception and demonstrated onboard detection of unexpected faults but the approach mainly relied on predefined hard thresholds and struggled with some advanced scenarios. Zhang et al. [5] developed a rudder fault detection scheme using RNNs with self adapting thresholds, showing high accuracy in actual trials in sea , but the approach required significant offline training at first an lacked real time adaptability. Wu et al. [6] introduced an unsupervised BiGAN-based anomaly detector for underwater gliders, which outperformed classical methods across multiple deployments consistently . While effective, their approach still depended on offline centralized training ,not addressing security concerns and using decentralization techniques ultimately limiting adaptability during live unsupervised missions.

Federated Learning for IoT and Edge Devices: Federated Learning (FL) offers privacy-preserving distributed model training without sharing of any raw data, making it an important consideration for AUV swarms in practical deployments . Blanco-Justicia et al. [7] conducted an in-depth study of FL security and privacy challenges, highlighting vulnerabilities such as poisoning attacks and gradient leakage. Li et al. [8] provided a survey on Federated learning applications in resource constrained IoT devices , emphasizing problems like heterogeneity of data gathered , low com-

putational power to process and locally train models –some of which can be ignored for AUVs. Wu et al. [9] proposed FedKD, a communication-efficient FL approach based on mutual knowledge distillation, which achieved accuracy close to FedAvg while reducing communication costs. However, the design still assumed availability of relatively large models not always feasible on constrained underwater vehicles.

Blockchain for Cybersecurity: Blockchain has emerged as a tool to enhance integrity in distributed ML systems. Zhang et al. [10] emphasized that lightweight blockchain integration can help detect tampering with FL model updates without adding significant overhead, though practical deployments remain limited. More recently, Chen et al. [11] proposed FedTKD, a blockchain-assisted FL framework that used logit-based verification to defend against malicious clients, demonstrating trustworthy aggregation in heterogeneous environments. While promising, blockchain designs still face trade-offs in scalability versus efficiency, making lightweight solutions critical for real-time AUV operations.

Knowledge Distillation in Federated Settings: Compressing learning patterns of larger models into smaller, lighter student models for use on edge devices is a common application of knowledge distillation . A thorough analysis of KD techniques was provided by Gou et al. [12], who discovered that they were very successful on devices with limited resources. Wu et al. [9] used FedKD to extend KD to federated learning, showing lower communication costs without sacrificing model performance. By aligning smaller student models across different clients, Chen et al. [11] conducted additional research and demonstrated how KD can improve functionality in heterogeneous FL. But as Gou et al. [12] pointed out, the majority of KD techniques rely on the availability of sizable, excellent teacher models, which can be difficult in AUV environments with limited shared datasets and restricted connectivity.

3 | PROPOSED FRAMEWORK: ARCHITECTURE AND METHODOLOGY

In this study , With a blockchain network serving as a decentralized trust layer, the suggested system is structured as a hierarchical framework that judiciously divides computational tasks between the resource-constrained AUV swarm and a potent central server.

3.1 | System Architecture

The architecture consists of three core components, as illustrated in Figure 2.

- **AUV Swarm (FL Clients):** This includes a field-based fleet of AUVs. In the federated learning process, every AUV serves as a client. It uses a lightweight "student" model for on-device anomaly detection in real time and has sensors to gather operational data. Each AUV trains its local model during training rounds, calculates a SHA-256 hash of the model weights that are produced, and sends the weights and hash to the Aggregation Server.

3.1.1 | Federated Training Protocol

For applications in AUVs in low computational power environments, a lightweight blockchain algorithm was integrated with a Federated Averaging (FedAvg) algorithm.

At each global round t, the central aggregator broadcasts the current model parameters w_t to a subset of participating clients. Next, each client k trains a local model on its private dataset D_k using stochastic gradient descent (SGD). Our implementation used a logistic regression style model (`SGDClassifier`) as the local model. It was trained with incremental updates (`partial_fit`) and had a loss set to `log_loss`. Each client created a SHA-256 hash

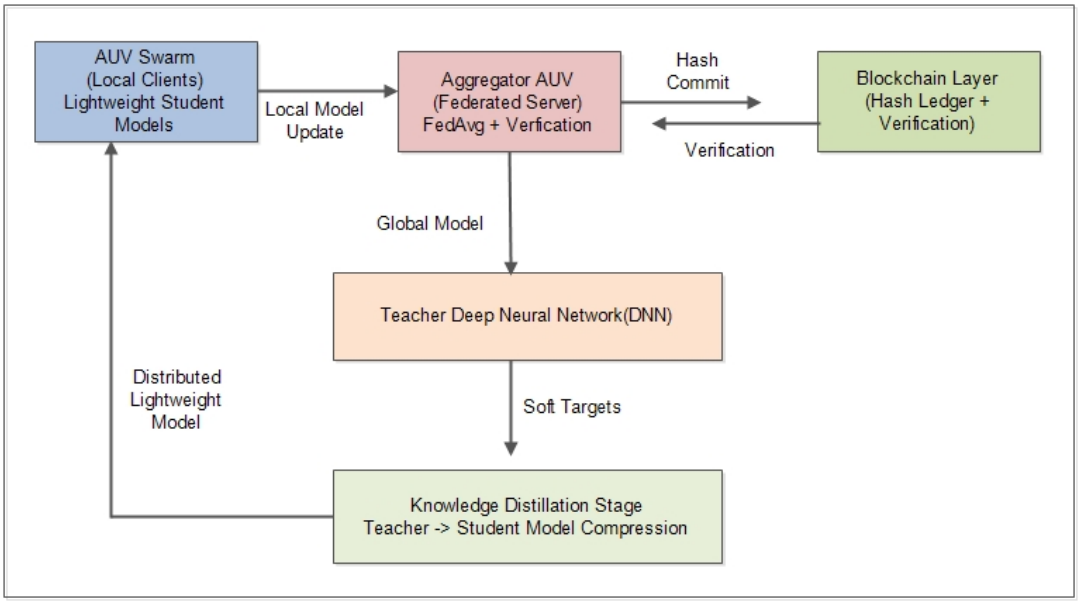

FIGURE 2 High-level architecture of the proposed hierarchical learning and security framework.

h_k = SHA256w_{t1}^k and updated weights w_{t1}^k, which were then stored on the blockchain before being transmitted. Only legitimate contributions were combined using the weighted FedAvg rule after the aggregator checked received updates against their registered hashes:

$$w_{t1} = \frac{{}_k\, n_k \cdot w_{t1}^k}{{}_k\, n_k},$$

where n_k denotes the number of local samples at client k. This blockchain-based verification ensured tamper resistance and accountability in potentially unreliable network conditions.

Algorithm 1 Federated Training with Blockchain Verification

Server initializes global weights w_0
for each round $t = 1, \ldots, T$ **do**
  Server selects subset of clients $\mathcal{K}_t$
  for each client $k \in \mathcal{K}_t$ **in parallel do**
    Receive w_t and train local SGD model on D_k
    Compute update w_{t1}^k, hash $h_k \leftarrow$ SHA256w_{t1}^k
    Commit h_k to blockchain ledger
    Send w_{t1}^k, h_k to server
  end for
  Server verifies hashes h_k against received updates
  Aggregate verified updates to form w_{t1}
end for

3.1.2 | Implementation Details

The `Flower` framework with `scikit-learn`'s `SGDClassifier` was used to implement the federated training. Standardized numerical features and one-hot encoded categorical features were employed in a batch-incremental training setup for every client. The **Smart Ocean Dataset** was used for training after being preprocessed and enlarged using the classifier-disagreement based resampling (CDBR) technique outlined in Section 4.1.

The main hyperparameters of our federated setup are summarized below:

- **Model:** SGDClassifier (`log_loss`)
- **Dataset:** Smart Ocean Dataset (expanded via CDBR)
- **Rounds:** 100
- **Local optimizer:** SGD
- **Local epochs:** 1 (partial fit per round)
- **Batch size:** Full local dataset
- **Learning rate:** Default (sklearn adaptive schedule)
- **Blockchain layer:** SHA-256 hashing of updates, ledger verification
- **Baseline accuracy:** 45.6%

3.1.3 | Aggregation Server (Aggregator AUV and Base Teacher)

There is *no constant command-and-control surface vessel* during deployment in the suggested framework. Rather, one AUV in the swarm is assigned the role of the **Aggregator AUV**, which is in charge of gathering and merging model updates at every training cycle. After training its local model, each participating AUV calculates a SHA-256 hash of its updated parameters and sends the hash and parameters to the Aggregator AUV. By temporarily storing hashes, the Aggregator AUV verifies integrity in real time. It then uses the Federated Averaging (FedAvg) algorithm to update the global model that the swarm has access to. All updates and hashes are added to the base station's blockchain ledger upon mission completion, guaranteeing long-lasting, impenetrable verification.

A **teacher–student knowledge distillation stage** is introduced after the global model has been consolidated and validated at base. Here, full GPU resources are used to train a high-capacity *Teacher Deep Neural Network (DNN)* on an enlarged Smart Ocean Dataset. Lightweight AUV models are unable to capture fine-grained patterns, but this model does. After that, the skilled instructor transfers softened class distributions to enhance generalization and robustness by distilling knowledge into the aggregated global model (the student).

Teacher Model Training: The teacher was implemented as a multi-layer deep classifier (512–256–128–64 hidden units with LayerNorm, LeakyReLU, and dropout). It was trained end-to-end on the expanded Smart Ocean dataset, using the following configuration:

- **Optimizer:** AdamW with cosine annealing LR schedule
- **Epochs:** 40 (with early stopping patience = 5)
- **Batch size:** 128,000 (full GPU memory)
- **Loss:** Cross-entropy
- **Precision:** Mixed-precision training (AMP)

TABLE 2 Teacher Deep Model Training Configuration

Parameter	Value
Architecture	[512, 256, 128, 64] + LayerNorm + Dropout
Optimizer	AdamW
Scheduler	Cosine Annealing
Batch size	128,000
Epochs	40 (early stop=5)
Loss	Cross-entropy
Precision	Mixed FP16/32 (AMP)
Best Val Accuracy	(to be reported from logs)

Algorithm 2 Teacher Model Training on Expanded Smart Ocean Dataset

Input: Expanded Smart Ocean dataset D_{exp}, teacher model T, learning rate η, epochs E
Initialize parameters θ_T of teacher model T
for e = 1 to E **do**
  for each batch $x, y \in D_{exp}$ **do**
    $y_{pred} \leftarrow Tx; \theta_T$
    Compute loss $\mathcal{L} \leftarrow$ CrossEntropyy_{pred}, y
    Update $\theta_T \leftarrow \theta_T - \eta \nabla_{\theta_T} \mathcal{L}$
  end for
end for
Output: Trained teacher model T

Knowledge Distillation Stage:

After the teacher model has been fully trained on the expanded Smart Ocean dataset, a compact student model — either a logistic regression–style classifier or a shallow neural network — is trained to mimic the teacher's behavior. The student receives the soft targets from the teacher and optimizes a combined loss that balances fidelity to ground-truth labels with alignment to the teacher's predictions.

The total KD loss is:

$$\mathcal{L}_{KD} = 1 - \alpha \cdot \mathcal{L}_{CE} y_s, y \quad \alpha \cdot T^2 \cdot KL\big(\sigma z_s T \,\|\, \sigma z_T T\big)$$

where:

- y_s — student predictions
- y — ground-truth labels
- z_s, z_T — logits of the student and teacher models

- T — softmax temperature
- α — weight balancing the hard and soft losses
- $\sigma\cdot$ — softmax function

TABLE 3 Knowledge Distillation Setup

Parameter	Value
Teacher	Deep DNN (512–256–128–64)
Student	Lightweight model (logistic regression style)
Optimizer	AdamW
Epochs	50
Batch size	8192
Temperature (T)	2.0
KD weight (α)	0.7
Dataset	Balanced Smart Ocean

Algorithm 3 Knowledge Distillation from Teacher to Student

Input: Trained teacher model T, student model S, dataset D, temperature τ, distillation weight α, epochs E
Initialize parameters θ_S of student model S
for e = 1 to E **do**
  for each batch $x, y \in D$ **do**
    $y_{teacher} \leftarrow \text{Softmax}Tx\tau$
    $y_{student} \leftarrow \text{Softmax}Sx\tau$
    Compute distillation loss $\mathcal{L}_{KD} \leftarrow \text{KLDiv}y_{student}, y_{teacher}$
    Compute hard loss $\mathcal{L}_{CE} \leftarrow \text{CrossEntropy}Sx, y$
    Total loss: $\mathcal{L} \leftarrow \alpha \cdot \mathcal{L}_{CE}\ 1 - \alpha \cdot \mathcal{L}_{KD}$
    Update $\theta_S \leftarrow \theta_S - \eta\nabla_{\theta_S}\mathcal{L}$
  end for
end for
Output: Trained student model S

3.2 | Blockchain-Secured Federated Learning

In order to guarantee safe and impenetrable model aggregation, the collaborative framework's core algorithm, Federated Averaging (FedAvg), is enhanced with a blockchain-based verification layer. Because autonomous missions require AUVs to work together without a constant supervisor and are subjected to potentially unstable communication environments, this design is essential.

One AUV is dynamically designated as the *aggregator* during the mission, acting as the temporary server in the suggested configuration. In addition to ensuring that the learning process stays resilient and fully distributed, this

eliminates the need for a surface vessel or coordinator based on land.

Each communication round unfolds as follows:

1. **Distribution:** The aggregator broadcasts the current global model weights w_t to all N participating client AUV ends.
2. **Local Training:** Each client k trains the model locally on its private dataset D_k, collected during its mission, which yields updated local weights w_{t1}^k.
3. **Hashing and Transmission:** To guarantee update integrity, each client genrates and applies a cryptographic hash $h_k = Hw_{t1}^k$ over its updated parameters. The tuple w_{t1}^k, h_k is transmitted to the aggregator, while h_k is registered on the blockchain logs shared among AUVs to verify each update.
4. **Verification and Aggregation:** Upon receiving updates, the aggregator re-computes $h'_k = Hw_{t1}^k$ and cross-checks it with the blockchain entry h_k. If verification passes, the update is included in aggregation; otherwise, it is discarded. Verified updates are combined using a weighted average:

$$w_{t1} = \sum_{k=1}^{N} \frac{n_k}{n} w_{t1}^k, \quad \text{where } n = \sum_{k=1}^{N} n_k.$$

This approach to the steps described above prevents any data or model poisoning and tampering attempts , as any mismatch between the shared parameters by each AUV and their blockchain -registered hashes invalidates the contribution by that AUV in the global model update. In effect , the blockchain implementation provides a decentralized trust mechanism that works well with the statistical robustness of FedAvg.

TABLE 4 Notation used in Blockchain-Secured Federated Learning

Symbol	Description
N	Total number of AUV clients
k	Index of a client
D_k	Local dataset of client k
n_k	Size of dataset D_k
w_t	Global model weights at round t
w_{t1}^k	Local model weights of client k after training in round t
$H\cdot$	SHA-256 cryptographic hash function
$\mathcal{L}_{KD}$	Knowledge Distillation loss
$\mathcal{L}_{CE}$	Cross-Entropy loss
z_s, z_T	Logits from student and teacher models
y	Ground-truth labels
T	Temperature parameter for KD
α	Weighting factor for KD loss

3.3 | Knowledge Distillation for Performance Enhancement

Although reliable model aggregation is guaranteed by blockchain-secured federated learning, resource-constrained AUVs may still find the final global model computationally demanding. We tackle this by implementing a post-training *Knowledge Distillation* phase, in which a compact student model absorbs the knowledge of a large teacher network.

Algorithm 4 Blockchain-Secured Federated Learning Round

Aggregator executes:
Initialize w_0
for $t = 0, 1, \ldots$ **do**
  $w_{updates} \leftarrow$
  for each client k in parallel **do**
    $w_{t1}^k, h_k \leftarrow$ ClientUpdatek, w_t
    if Verifyw_{t1}^k, h_k = true **then**
      Append w_{t1}^k, n_k to $w_{updates}$
    end if
  end for
  $w_{t1} \leftarrow \frac{n_k}{n} w_{t1}^k$
end for

procedure ClientUpdatek, w_t
  $w_{t1}^k \leftarrow$ Trainw_t, D_k
  $h_k \leftarrow H w_{t1}^k$
  Register h_k on blockchain
  return w_{t1}^k, h_k

While the student (such as logistic regression or a shallow MLP) is optimized to balance fidelity to both the ground-truth labels and the teacher's predictions, the teacher is trained on the larger Smart Ocean dataset. This makes it deployable on embedded systems and enables the student to approximate teacher-level accuracy with much less complexity.

The total KD loss is defined as:

$$\mathcal{L}_{KD} = 1 - \alpha \cdot \mathcal{L}_{CE} y_s, y \quad \alpha T^2 \cdot KL\left(\sigma \frac{z_s}{T} \parallel \sigma \frac{z_T}{T}\right)$$

where y_s denotes the student predictions, y the ground-truth labels, z_s, z_T are the logits of student and teacher, T the softmax temperature, α the distillation weight, and $\sigma\cdot$ the softmax function.

The federated knowledge is compressed into a lightweight form while maintaining robustness during this distillation stage. Thus, the student model satisfies the deployment requirements of distributed AUV networks by striking a balance between accuracy, efficiency, and security.

4 | IMPLEMENTATION AND RESULTS

4.1 | Dataset and Preprocessing

The Dataset used in this experiment , specifically SFI Smart Ocean Dataset from IEEE Data Port ,a multi sensor dateset with appropriate labels being used [13]. However , the small size of the dataset limited the learning capabilities for decision boundaries of our Deep Neural model used in the experiment being used as the teacher. To solve this problem , to enlarge the dataset to a size suitable or the deep learning model , we used Classifier-Disagreement

Based Resampling (CDBR) approach which enlarges the dataset while preserving anomaly features [14]. This method involves training a baseline classifier on the original dataset, comparing predictions with ground truth, and selectively resampling both correctly classified (stable) and misclassified (hard-case) samples. By using this , an enhanced dataset of roughly 1 million rows was generated that were strictly balanced distribution of 50% of normal and 50% anomalous samples.. Salinity (ppt), temperature (°C), depth (m), and sound speed (m/s) are the four continuous features that were taken from the raw sensor logs and used as inputs for anomaly detection. Since date, time, and identifier metadata fields don't directly support anomaly discrimination, they were left out. All features were standardized using z-score normalization before the model was trained:

$$x' = \frac{x - \mu}{\sigma}$$

Where μ and σ represent the mean and standard deviation of each feature determined on the training split.

4.2 | Simulation Environment

Experiments were conducted in a Google Colab environment utilizing an NVIDIA A100 GPU. The federated learning simulation was implemented using the Flower framework with PyTorch. The teacher model and Knowledge Distillation process were also implemented in PyTorch. The blockchain component was simulated in Python using the SHA-256 hashing algorithm. The key parameters for the KD experiment are listed in Table 2.

TABLE 5 Simulation Parameters for Knowledge Distillation

Parameter	Value
Environment	Google Colab
GPU	NVIDIA A100
Framework	PyTorch
Optimizer	SGD
Batch Size	8192
Learning Rate	1×10^{-3}
Epochs	50
KD Alpha (α)	0.7
KD Temperature (T)	2.0

4.3 | Performance Evaluation

The baseline student model, representing the outcome of a standard federated learning process, was found to be ineffective for the anomaly detection task, achieving a validation accuracy of only 45.8%. The model was then subjected to 50 epochs of Knowledge Distillation, with performance detailed in Table 6.

TABLE 6 Student Model Accuracy during Knowledge Distillation

Phase	Validation Loss	Validation Accuracy
Pre-KD (Baseline)	0.7167	45.8%
Epoch 1	0.6343	57.6%
Epoch 10	0.6064	61.2%
Epoch 20	0.5999	61.5%
Epoch 30	0.5984	62.1%
Epoch 40	0.5954	62.5%
Epoch 50	0.5941	62.9%

4.4 | Performance Metrics

Three Level of through analyses were conducted by us to to confirm and judge the effectiveness of the proposed structure . First being the baseline student performance prior to doing KD , second being the logs and dynamics during KD such as number of epochs and improvement. A third level was used to judge post-distillation outcomes for comparison to the baseline model. All the metrics were recorded through training cycles and epochs to ensure reproducibility of the experiment if needed. Visualization is included through the result presentation section.

4.4.1 | Baseline Model Performance

On the anomaly detection task, the baseline student model—which was based on conventional federated learning without any modifications—performed poorly. It only obtained **45.8%** validation accuracy, as indicated in Table 7, which is less than the expected level of random chance on a balanced binary dataset. This highlights two drawbacks: (i) lightweight student models when trained alone do not have enough representational capacity, and (ii) this shortcoming is exacerbated by federated data, which is by nature distributed and noisy

Figure 3 illustrates the unstable trajectory of validation accuracy and loss during baseline training, underscoring the inadequacy of the naive federated model. This justified the need for Knowledge Distillation.

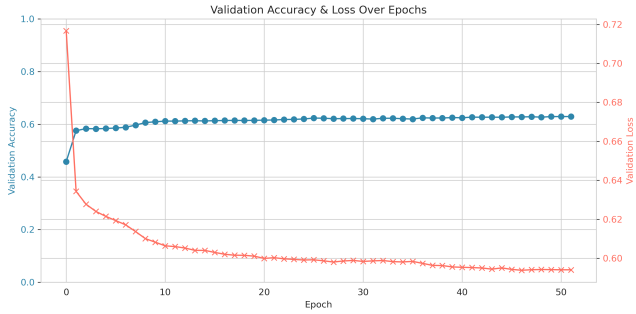

FIGURE 3 Baseline model performance: validation accuracy and loss across training epochs. The low accuracy and unstable curve highlight poor anomaly learning capability.

4.4.2 | Knowledge Distillation Training Dynamics

The baseline model was refined using **50 epochs of Knowledge Distillation**. KD enables the student to approximate the richer decision boundaries of the teacher model by learning both probability distributions and inter-class similarities. The trajectory of validation accuracy and loss is visualized in Figure 4. The blue curve (accuracy) rises sharply in the first 10–15 epochs before plateauing around 62–63%, while the orange curve (loss) shows a steep decline followed by stabilization. Together, these curves demonstrate rapid assimilation of teacher knowledge and stable convergence without overfitting.

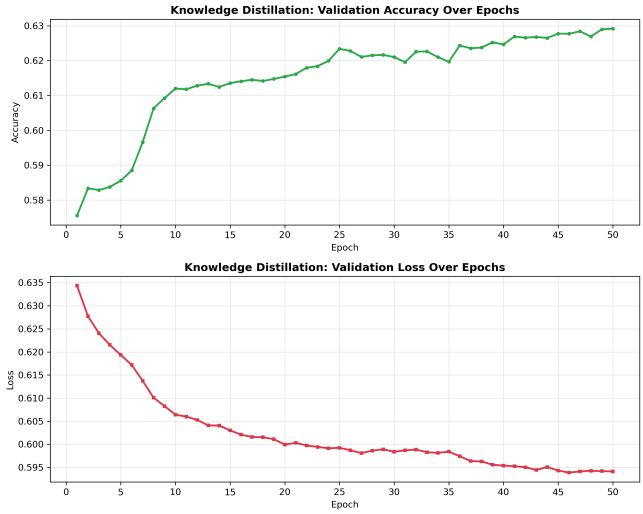

FIGURE 4 Validation accuracy and loss of the student model over 50 epochs of Knowledge Distillation. Early epochs show steep improvements, followed by stable convergence.

4.4.3 | Federated Training Round Analysis

In parallel, we examined fifty communication rounds' worth of federated training metrics, as illustrated in Figure 5. Over the course of these rounds, the results demonstrate steady and progressive improvements across multiple performance indicators, including accuracy, precision, recall, F1-score, and Receiver Operating Characteristic and Area Under the Curve (ROC-AUC). The gradual yet consistent enhancement in validation accuracy, reaching a value of 66.9%, indicates that the federated aggregation process continues to refine the global model, improving its generalization capabilities prior to the application of Knowledge Distillation . Figure 6 presents a consolidated view of both accuracy and F1-score across the communication rounds. The tandem improvement of these two metrics provides strong evidence that the observed performance gains are not the result of bias toward any particular class. Instead, they reflect a well-balanced and robust anomaly detection capability across the dataset. Moreover, the simultaneous rise in multiple evaluation metrics underscores the effectiveness of federated learning in harmonizing local updates from heterogeneous clients, leading to a more resilient and accurate global model

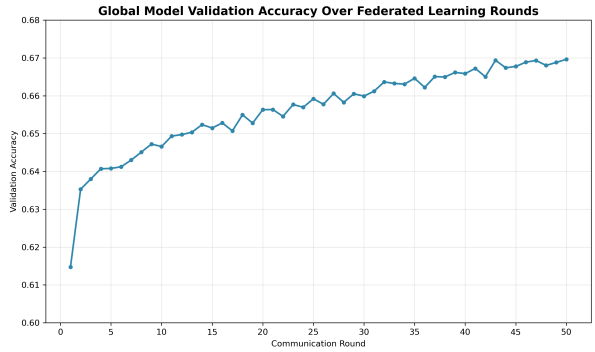

FIGURE 5 Global validation accuracy over 50 federated training rounds, showing consistent performance gains.

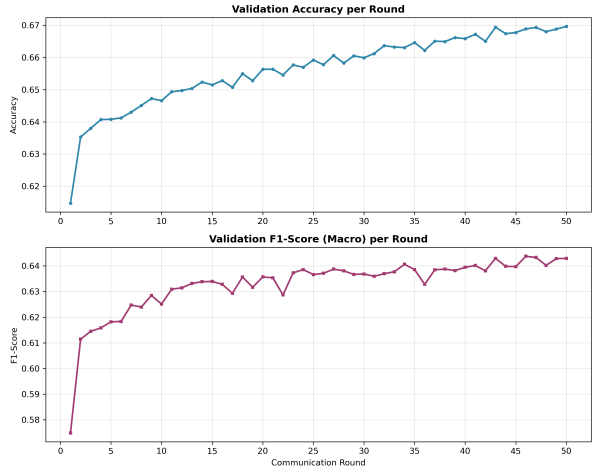

FIGURE 6 Combined trajectory of validation accuracy and F1-score across federated rounds, reflecting balanced improvements.

4.4.4 | Performance Improvement Through KD

Figures 7 and 8 provide a detailed and comprehensive comparison of model performance metrics both prior to and following the application of Knowledge Distillation . As observed in these figures, the validation accuracy demonstrated a substantial increase, achieving a relative gain of 37.4%, rising from an initial 45.8% to a notably higher 62.9%. This marked improvement in validation accuracy clearly indicates that the model benefited significantly from the KD process, reflecting enhanced learning efficiency and more robust feature extraction capabilities. In parallel, the validation loss exhibited a meaningful decrease, declining from 0.717 to 0.594. This reduction in loss not only signifies improved alignment between the predicted and true labels but also suggests that the model's internal representations became more stable and reliable after the distillation process. Taken together, these results illustrate that Knowledge Distillation positively influenced both the model's predictive performance and its confidence in making accurate predictions.

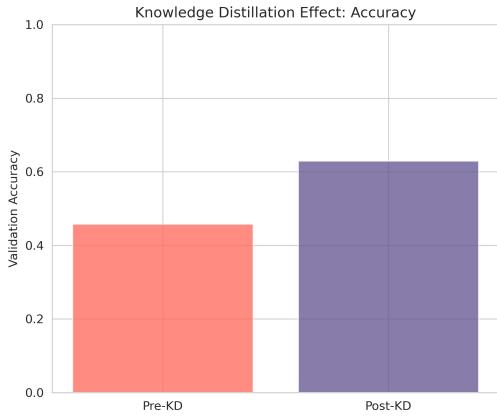

FIGURE 7 Validation accuracy before (Pre-KD) and after (Post-KD) Knowledge Distillation. Accuracy increased from 45.8% to 62.9%.

FIGURE 8 Validation loss before (Pre-KD) and after (Post-KD). Loss reduced from 0.717 to 0.594.

4.4.5 | Epoch-Wise Accuracy Gains

A fine-grained view of accuracy progression across KD epochs is presented in Figure 9. One epoch is represented by each step, and the monotonic increase shows consistent and dependable gains. This graphic supports the idea that advancements were the result of steady knowledge transfer rather than sporadic variations.

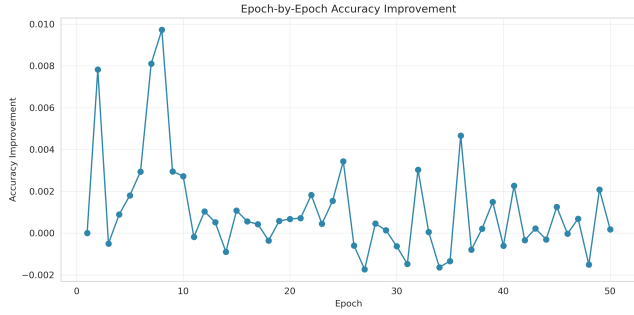

FIGURE 9 Epoch-wise accuracy improvements during Knowledge Distillation, showing consistent upward progression.

4.4.6 | Final Evaluation Metrics

A comprehensive evaluation of the distilled model is presented in Table 7. Beyond accuracy, all key metrics improved substantially. For example, Recall improved from 42.9% to 63.6%, demonstrating enhanced sensitivity to anomalies, while Precision improved from 45.7% to 62.8%, reducing false alarms. The consistent improvements across Accuracy, Precision, Recall, F1-score, and Specificity highlight that KD produced a balanced and reliable anomaly detector.

TABLE 7 Comprehensive Metrics Pre- and Post-Knowledge Distillation

Metric	Baseline Model (Pre-KD)	Final Model (Post-KD)
Accuracy	45.8%	62.9%
Precision	45.7%	62.8%
Recall	42.9%	63.6%
F1-Score	44.2%	63.2%
Specificity	48.8%	62.3%

4.5 | Discussion

The results of the study emphasize and address the synchronization issue between privacy preserving federated learning and model performance on resource-constrained platforms. The baseline student model ensures decentralization of the training and logging process ,its accuracy of 45.8% was insufficient for anomaly detection in AUVs. This highlighted the challenges current security setups face which aim to enhance privacy of a network.

Knowledge Distillation provides an effective approach to address these challenges. By transferring knowledge from the teacher model, the student's accuracy improved to 62.9%, pushing an overall gain of 37.4% in accuracy on the same student model.Importantly, this improvement was achieved without altering the student model's architecture at all , making KD a suitable and desirable solution for environments with strict computational and energy limits. The integration of Blockchain added transparency and security of the training process.

5 | CONCLUSION AND FUTURE WORK

Our work proposes an hierarchical framework that integrates federated learning secured via blockchain integration with knowledge distillation specifically designed for AUV systems. In this proposed structure FL allows multiple AUVs to collaboratively train a global model without sharing raw data, preserving privacy, while KD transfers knowledge from a complex teacher model to a simpler student model, increasing its raw accuracy on the same lightweight model, providing a system for low scale hardware to work close to a larger scale model .

Going Further in future , several studies and experiments can be pursued to further validate and extend this approach :

- Physical deployment: Testing the framework on actual AUV hardware to evaluate real-world performance, latency, and resource consumption, ensuring that the models can operate efficiently in constrained underwater environments.
- Lightweight blockchain mechanisms: Investigating alternative blockchain consensus protocols, such as Proof-of-Authority, to reduce computational and communication overhead while maintaining security and integrity of the decentralized learning process.
- Broader applicability: Extending the framework beyond AUVs to other constrained cyber-physical systems, including drones, IoT networks, and mobile edge devices, to explore its generalizability for secure and private anomaly detection in diverse domains.

In summary, the proposed framework demonstrates that FL, KD, and blockchain can together support secure, private, and efficient anomaly detection on edge devices.

references

[1] Pang G, Shen C, Cao L, Hengel AVD. Deep learning for anomaly detection: A review. ACM computing surveys (CSUR) 2021;54(2):1–38.

[2] Wang K, Shen C, Li X, Lu J. Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications. IEEE Transactions on Intelligent Transportation Systems 2025;.

[3] Gou J, Yu B, Maybank SJ, Tao D. Knowledge distillation: A survey. International journal of computer vision 2021;129(6):1789–1819.

[4] Raanan B, Bellingham JG, Zhang Y, Kemp M, Kieft B, Singh H, et al. Detection of unanticipated faults for autonomous underwater vehicles using online topic models. Robotica 2018;36(10):1521–1538.

[5] Zhang Z, Zhang X, Yan T, Gao S, Yu Z. Data-Driven Fault Detection of AUV Rudder System: A Mixture Model Approach. Machines 2023;11(5):551.

[6] Wu P, Harris CA, Salavasidis G, Lorenzo-Lopez A, Kamarudzaman I, Phillips AB, et al. Unsupervised anomaly detection for underwater gliders using generative adversarial networks. Engineering Applications of Artificial Intelligence 2021;102:104379.

[7] Blanco-Justicia A, et al. Privacy-preserving federated learning for anomaly detection in cyber-physical systems. Engineering Applications of Artificial Intelligence 2021;106:104468.

[8] Li D, Han D, Weng TH, Zheng Z, Li H, Liu H, et al. Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Computing 2021;25:10613–10632.

[9] Wu Z, et al. Communication-efficient federated distillation learning via mutual knowledge transfer. Nature Communications 2022;13:2030.

[10] Zhang Y, et al. Blockchain-empowered federated learning: a survey. ACM Computing Surveys 2023;.

[11] Chen L, et al. FedTKD: A Trustworthy Heterogeneous Federated Learning Based on Adaptive Knowledge Distillation. Entropy 2024;26(1):96.

[12] Gou J, Yu B, Maybank SJ, Tao D. Knowledge distillation: A survey. International Journal of Computer Vision 2021;129:1789–1819.

[13] van Walree P, Otnes R, Tomasi B, Henriksen B, Danre JB, Øivind Bergh, SFI Smart Ocean dataset for underwater acoustic communications. IEEE Dataport; 2025. https://dx.doi.org/10.21227/3aa6-4k33.

[14] Carvalho M, Pinho AJ, Brás S. Resampling approaches to handle class imbalance: a review from a data perspective. Journal of Big Data 2025;12(1):71.