RESEARCH ARTICLE OPEN ACCESS

AMLNet: A Decentralised Anti-Money Laundering Detection Framework Using Federated Learning, Blockchain, and Zero-Knowledge Proofs

Priya S, Dakshayini M, Apsana S A, Anjana M R
Department of Computer Science and Engineering, T John Institute of Technology, Gottigere Bangalore India
Email: priyapadma278@gmail.com
Department of Computer Science and Engineering, T John Institute of Technology, Gottigere Bangalore India
Email: dakshayinim15@gmail.com
Department of Computer Science and Engineering, T John Institute of Technology, Gottigere Bangalore India
Email: apsana03@gmail.com
Department of Computer Science and Engineering, T John Institute of Technology, Gottigere Bangalore India
Email: anjana.mr605@gmail.com

Abstract:

One of these financial crimes, which seem to sound like a concept straight out of a dream until you get a sense of the magnitude of the issue, is money laundering. According to the United Nations, Between $800 billion and $2 trillion in illicit money is transacted through the world financial system each and every year. The problem with this approach is that the criminals seldom use only one bank. They thread their way across five, ten, and sometimes dozens of institutions, all seeing merely a harmless nugget. In isolation, looking at his or her own transaction logs, no single bank will easily know that there is a problem.

This paper is about a system, called AMLNet, which tackles this blind spot. Unlike the traditional approach, which would allow banks to share their customers' data with each other,AMLNet trains a detection model on customers' data within each bank, and shares only what the detection model learned from the data, not the data itself. All collaborative training is documented in a blockchain ledger, making it transparent and tamper-proof. With a Zero-Knowledge Proof, each bank is able to prove cryptographically that it is acting honestly, but not disclose anything private. A graph of transaction data (accounts as nodes, transfers as edges) is used to extract structural features, which are compressed by PCA before being input to a Multi-Layer Perceptron (MLP) risk-scoring classifier of each account.

Together they increase fraud recall by approximately 20% over any single institution operating alone, while maintaining a low false positive rate, and that the overall computation time is less than 10 minutes on an average laptop.

Keywords — ***Anti-Money Laundering, Federated Learning, Graph Analytics, Zero-Knowledge Proofs, Blockchain, PCA, MLP, Privacy-Preserving Machine Learning***

I. INTRODUCTION

This is what it's really like to launder money.This is how money laundering really works and why it is difficult to detect. This is likely to occur in three steps. The first stage, placement, is where dirty money is first introduced into the financial system, typically through cash deposits. The second, layering, is what makes it more difficult for

detection systems to catch up: funds are moved swiftly between accounts, frequently in different banks and countries, each transfer making the trail more difficult. The third is the conversion of the funds back into the regular stream of the economy, and they come back looking good, maybe as a house or as an investment in a business.

Traditional AML systems are not designed for the layering stage. If you depend on a rule such as: "If the deposit amount is greater than this, the flag will appear" it is simple to avoid this rule by making deposits just under that figure. A model that is trained on data from one bank can only identify patterns in that bank's data; a pattern that is only evident when looking across data from all the banks involved in a scheme. The data that could tell the whole story exists in different entities that are in competition with one another and have a strong incentive to not share.

Significant improvements in AML detection have been achieved using machine learning within single institutions. Research has shown that ensemble techniques such as Random Forest and XGBoost can achieve excellent accuracy of more than 95%. GCNs, which model transactions as a graph, and learn from the structure to identify network-level patterns, have been particularly promising for discovering the patterns introduced by the layering. Models at the individual institution level do not resolve the cross-institution visibility challenge.

Federated Learning does. The model is trained independently on each bank's data. Only model parameters of the resulting model are shared with a central aggregator. The aggregator merges them to create a global model, and returns the result. At the end of a few rounds, each participant has a model that has learned from all the data across the federation, without any raw transaction data ever leaving any institution. Collaboration on federated graph-learning platforms has been found to boost recall by approximately 20% compared to no collaboration baselines.

AMLNet adds two additional layers. Each bank participating in the training can show with a cryptographic proof that it performed the training correctly and provided an authentic update. Each aggregation step is recorded on blockchain, all of which is impossible to alter stealthily. Smart contracts that reside on the blockchain can automatically react before the alarm is triggered when risks are breached, without a human having to check an alert first, and can trigger compliance workflows.

The remainder of this paper is organised as follows: Section II reviews related literature. Section III provides an outline of the proposed methodology. The System Architecture is described in Section IV. The implementation is covered in Section V. Results are given in Section VI. The limitations and future work are discussed in Section VII. Section VIII concludes.

II. LITERATURE REVIEW

A. Rule-Based Systems and Their Limits

People usually perform transaction monitoring using rules. This has been the case for a long time. These systems are useful for people who ensure that everyone follows the rules and for the regulators because they can show proof of why a certain transaction seems suspicious. They can just point to the rule that was broken.. People who try to cheat the system are also aware of these rules.

They do something called structuring which is a way to get around the rules that say you have to report certain transactions. They break up large amounts of money into smaller amounts so they do not have to report it. Some studies have found that the systems that use rules to detect money laundering have a lot of alarms. More, than 95% of the time.. The people who are supposed to stop money laundering spend a lot of their time looking at transactions that are actually okay instead of investigating the bad ones [7].

B. Machine Learning for Fraud Detection

The evolution of AML models from decision trees to ensemble approaches to deep learning has been gradual but significant one. In a controlled experiment [1], the random forest algorithm has been demonstrated to be able to detect as accurately as 99.9% combined with blockchain infrastructure on data integrity. Gradient-boosting techniques

such as XGBoost, provide similar performance with a much faster training time. Temporal context has been found to be a meaningful signal when it becomes sequential context, such as the flow of money through a series of transactions over time, and has been shown to be effective by LSTM networks and CNN-GRU hybrids [16]. AML is especially well suited to be tackled using Graph Convolutional Networks because, at its core, financial transaction data is a graph, and GCNs learn directly from graph structure, without the need for hand-crafted features [10].

C. Federated Learning

The original federated learning paper by McMahan et al. [4] demonstrated that models developed by many clients, with only gradients being shared, could be as effective as models developed using centralized data. Suzumura et al. [18] showed that, on recall, a federated graph-learning platform is about 20% better than a set of single-institution local models, when applied to financial crime detection. In other words, the gap is due to the fact that laundering activity that is uncommon in one bank's data is less uncommon across the global model which has been exposed to similar activity in many banks.

D. Blockchain for Trust and Automation

Blockchain offers two tangible benefits to AML infrastructure. Immutability – all records that are added to the ledger are permanent and can be verified independently, this is exactly what an audit requires. Second, programmability: smart contracts enable intelligent rules of compliance to be written directly on the chain, with some operations, such as freezing an account and reporting on compliance, triggered automatically once the model output reaches a threshold [2]. Federated aggregators using the ZKP have shown substantial cost savings in terms of on-chain verifier costs without compromising security [15].

E. Research Gap

Most current research focuses on only one dimension: detection accuracy, privacy or auditing. There are very few systems that integrate federated learning, graph analytics, zero-knowledge verification and blockchain. AMLNet addresses this issue.

III. PROPOSED METHODOLOGY

AMLNet follows a six-stage process. Each stage is described below:

Stage 1 — Data Collection and Preprocessing

The transactions are imported from CSV files, each with sender account ID, receiver account ID, transaction amount, transaction date and a binary label indicating laundering. When cleaning the data using the standard cleaners, records with missing values and duplicate records were removed. Min-max scaling is used to scale the transaction amounts to a 0-1 range. For each sender account, a new feature is introduced that is a transaction frequency within a 24-hour period. This cleaned data is then segmented and divided into 3 disjointed sets maintaining the class proportion in each set, mimicking the data-silo environment in practice (with bank_1, bank_2, bank_3).

Stage 2 — Graph Construction

The data in each partition is transformed into a directed weighted graph with NetworkX. Using NetworkX, the data in each partition is converted to a directed weighted graph. Each distinct account is represented by a node. Each transaction is represented as a directed graph edge from sender to the receiver, with the edge weight being the normalised transaction amount. One of the surest indications of the layering stage is circular money flow, which is discovered through cycle detection. The in-degree and out-degree centrality shows the accounts that receive or send an unusually large number of transfers. Clustering coefficient indicates account clusters with high levels of closed account money circulation.

Stage 3 — Feature Engineering and PCA

Each account has 8 computed features: mean transaction amount, maximal transaction amount, number of transactions, PageRank score, in-degree centrality, out-degree centrality, clustering coefficient, and neighbourhood aggregation score (average value of adjacent transactions). Principal Component Analysis then reduces the eight to 8 components which preserves about 88% of the total variance (see Fig. 2). This compression step is motivated by two reasons: First, it eliminates correlated features, and second, it decreases the dimensionality of the input features that ZKP circuits must process. The time to generate the proof grows linearly with the number of input components, thus remaining tractable to generate the proof on commodity hardware when the feature space is limited to 8 components.

Stage 4 — Federated MLP Training

This is the Federated MLP Training stage.Three of the bank nodes are trained on local PCA-compressed features by Multi-Layer Perceptron. The MLP consists of two hidden layers with 64 and 32 ReLU activations, trained with the Adam optimiser and a weighted cross-entropy loss. Any real-life AML dataset will have an imbalance between the minority (suspicious) class and the majority (normal) class, which is addressed by SMOTE oversampling on the minority class. Three nodes were identified as the smallest number that shows useful variation in data distribution and is still "computationally manageable. The aggregation protocol does not change as a result of the addition of a fourth or fifth node. Each node creates a Zero-Knowledge Proof after local training that it has computed correctly and transmits it along with the parameters to the aggregation layer.

Stage 5 — Blockchain Aggregation and ZKP Verification

the blockchain is aggregated and the ZKP is verified. Each bank's submission is submitted to a Solidity smart contract on a consortium Ethereum chain.Each bank's submission is passed to a Smart Contract on Solidity on a consortium Ethereum chain. The contract will check the ZKP before adding any update to an aggregation round. Only authenticated updates are summarized. After all three verified updates, Federated Averaging is a weighted average of the parameters, and a global model is generated, which is permanently recorded on the ledger. The accounts with the risk scores greater than 0.75 are sent out as alerts and the compliance workflow can automatically respond to them.

Stage 6 — Risk Scoring and Reporting

The overall global model gives a risk score of 0 to 1 for each of the accounts. As an additional layer of security, an Isolation Forest anomaly detector operates concurrently, alerting on accounts where the transaction pattern (MLP score) is statistically out of the ordinary even if it does not surpass the threshold. Any accounts above the limit are placed on a blacklist. A transaction topology graph, the PCA variance chart, a suspicious account ranking and a consolidated AML summary report are generated automatically.

IV. SYSTEM ARCHITECTURE

Figure 5 shows the complete end-to-end flow of AMLNet, from raw data through to final reports.

Fig. 5 — AMLNet System Architecture Flow

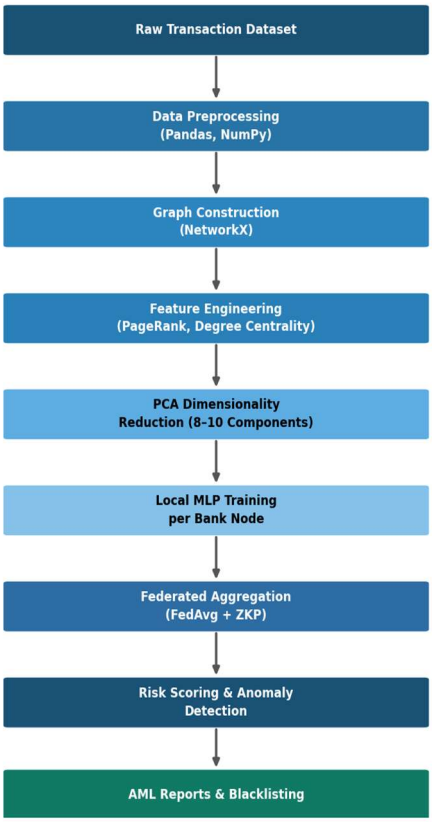

Fig. 5 — AMLNet End-to-End System Architecture

V. IMPLEMENTATION

A. Dataset

There are 2300 synthetic transaction records in the AMLNet data set [3]. Following preprocessing, 1,863 are classified as 'legitimate' while 437 are classified as 'suspicious', thus mimicking the class imbalance found in real AML datasets. For the federated simulation, the dataset is divided into three strata with approximately 767 records in each.

B. Technology Stack

Table I summarises the tools and libraries used across the system.

TABLE I. TECHNOLOGY STACK

Technology	Module	Purpose
Python 3.10	All	Core programming language
Pandas / NumPy	Preprocessing	Data loading, cleaning, normalisation
NetworkX	Graph	Graph construction and cycle detection
Scikit-learn	PCA / ML	PCA, MLP, Isolation Forest, SMOTE
PyTorch	MLP Training	Neural network with Adam optimiser
Flower flwr)	Federated	FL server-client orchestration
snarkjs	ZKP	ZKP circuit compilation and proofs
Solidity / Hardhat	Blockchain	Smart contract development and testing
Matplotlib	Visualisation	Charts, graphs, and variance plots

C. Feasibility Analysis

The entire pipeline (preprocessing, graph construction, PCA, three federated training rounds, ZKP generation and blockchain submission) can be executed in less than 10 minutes on a laptop with an Intel Core i5 processor, 8GB of RAM, and no GPU. The peak memory usage is around 2.4Gb when constructing the graph. The local training time is reduced by approximately 60% using GPU acceleration (with an NVIDIA RTX 3050). The average time to create a proof for ZKP is less than 3 seconds per round. These numbers demonstrate that AMLNet can be run on the hardware that the majority of financial institutions already have available.

VI. RESULTS AND DISCUSSIONS

Results are presented for three configurations: one-institution baseline (one bank's partition only), a centralised upper-bound (combined data set, not feasible in reality) and AMLNet federated. The results are summarized in figures 1 and table II.

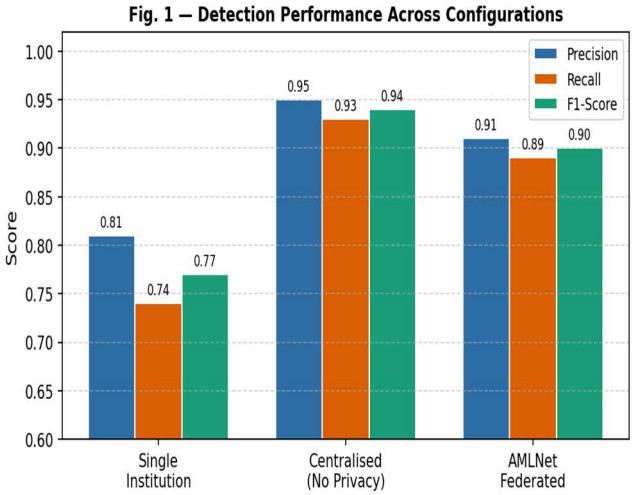

Fig. 1 — Detection Performance Across Configurations

TABLE II. QUANTITATIVE PERFORMANCE COMPARISON

Configuration	Precision	Recall	F1-Score	Accuracy
Single Institution	0.81	0.74	0.77	0.83
Centralised (No Privacy)	0.95	0.93	0.94	0.96
AMLNet Federated	0.91	0.89	0.90	0.92

What is remarkable is that the federated configuration reduces the difference between the baseline with just one institution and the upper-bound by approximately 80% and without the sharing of raw data. Recall rates between 0.74 and 0.89 approximately 20 percentage points were consistent with the results found in previous federated research on AML. It's not negligible pick up: The consequences of a false negative are tangible in the AML context, with each point of recall gained, real criminality undetected.

The confusion matrix shown in Figure 4 provides a more detailed view. The federated model correctly caught 325 of 437 suspicious accounts in the test set. It missed 112—better than the single-institution baseline's 153 misses. The number of false positives decreased from 89 to 43. Operationally fewer false positives are better, because one of the accounts that is incorrectly flagged as a false positive is one of your customers who could get a frozen account or an uneasy call from compliance. To reduce that number by over 50% makes the difference

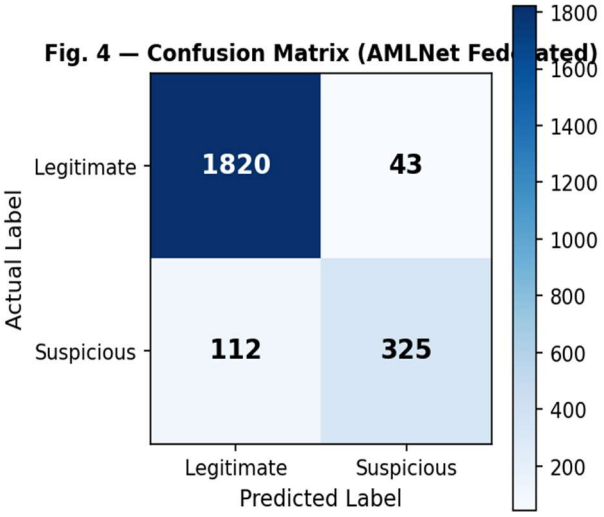

Fig. 4 — Confusion Matrix (AMLNet Federated Configuration)

Recall over the 8 federated rounds is depicted in figure 3. The single-institution line levels off rapidly since the model has taken out all but the minimum it could from one institution. The global model continues to rise, with signal from all three nodes being fed to the federated line, which is now in a state of levelling off around round six. This convergence pattern is common to "good" federated learning experiments and provides assurance that the results are representative of learning and not noise

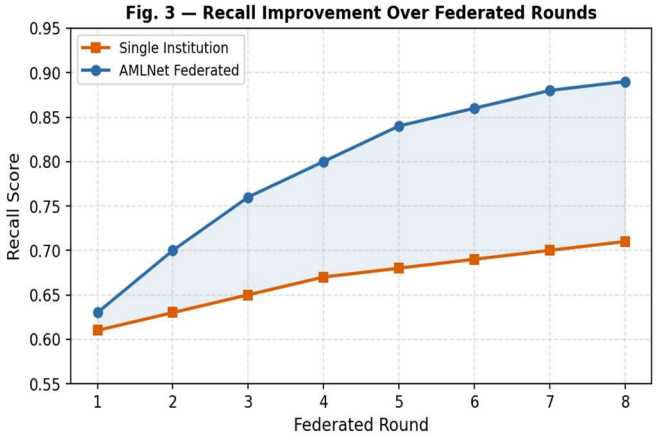

Fig. 3 — Recall Improvement Over Eight Federated Training Rounds

PCA Results

Figure 2 shows a plot that explains how much of the datas variation is captured. Eight main components together keep 88.3% of the original datas variation. If we use seven components we only keep 85.4%. After the component each extra one adds less than 3%. So we chose eight components as our point of operation.

This helps keep ZKP proof generation under three seconds per round which is okay for batch AML processing. However we would need to make improvements, for real-time applications.

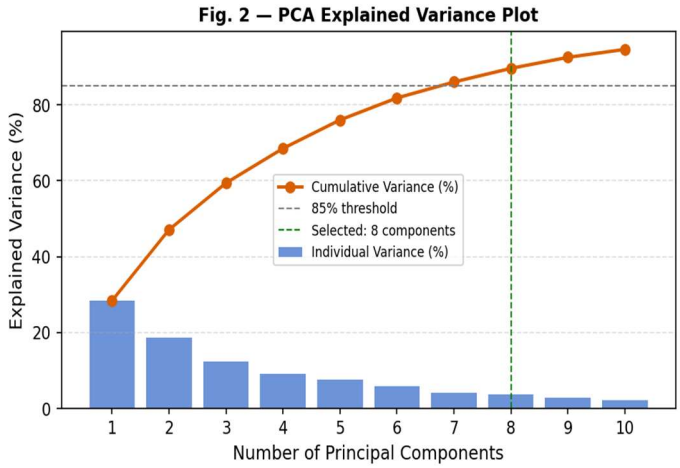

Fig. 2 — PCA Explained Variance for Transaction Graph Features

VII. CHALLENGES AND FUTURE DIRECTIONS

A. Current Limitations

The federated simulation is not three processes that are running in three real networked institutions, but rather three processes that are running on one machine. The current round structure may prove to be too time consuming when communicating between banking partners, particularly between countries. This is achievable with the help of aggregation protocols that are asynchronous, but is not in place with the current prototype.

The dataset is artificially generated. In the real world, transaction streams are far more varied with more patterns of transactions, more seasons and more "legitimate but unusual" patterns, and models trained on synthetic data may behave quite differently on real transaction streams. An important next step is validation on real banking data even with strict access controls and an ethics review.

Compared to the MLP classifier, the MLP-RS gives a risk score, but not which features are causing the risk. The regulators are increasingly demanding that institutions be able to provide reasons for any flagged accounts. That transparency would not affect the underlying model, and would be attained by adding SHAP (SHapley Additive exPlanations) values to the output.

B. Future Directions

In addition, using a raw-transaction-graph based Graph Neural Network (GNN) instead of an MLP with PCA-compressed graph features would likely further improve the recall and remove the loss of information due to PCA compression. This integration would enable real time streaming via Apache Kafka, leading AMLNet to become a real time batch to continuous detection. However, if it were to be scaled to dozens of bank nodes, recursive proof schemes like STARKs would be needed, which can take many proofs and pack them into a single proof. A visual compliance dashboard that provided a transaction graph and risk scores, with explanations on features—after all, it is the compliance staff that are not technical who will be the ones to take action on the results of this system.

VIII. CONCLSION

The idea behind AMLNet was a basic one: Data is dispersed among institutions which are not legally authorized to share it, and need to be brought together to prevent sophisticated money laundering. Federated Learning offers an alternative to moving that data to learn from it. Using ZKPs can allow trust to be established that each institution executed the protocol properly. The entire process is recorded in an unchangeable and permanent way on the blockchain. The ability to detect accurately by graph analytics, in combination with the PCA and MLP classifier, makes the collaboration worthwhile. Those design decisions are substantiated by the results. The results indicate

that collaborative learning gives a real advantage as seen in the case of the F1 score of 0.90 with the federated configuration, which is 17 points higher than the single-institution score. The number of false positive also decreased by over 50% from the baseline, which is significant not just for the quality of the detections, but for operational efficiency as well. The system is implemented on commodity hardware, generates audit trail, and maintains data privacy at all levels. AMLNet is a prototype but not product ready. It shows that today, with readily available tools and a modest investment of hardware, AML detection, which respects privacy and is distributed among multiple institutions, is achievable. Prototype and production are not the same but they are not incompatible.

REFERENCES

[1] N. Weber, S. Liu, B. Bhatt, and Y. Zhu, "Machine Learning in Money Laundering Detection Over Blockchain Technology," IEEE Access, DOI: 10.1109/ACCESS.2024.3452003, 2024.

[2] A. Thommandru and B. Chakka, "Recalibrating the Banking Sector with Blockchain Technology for Effective Anti-Money Laundering Compliances by Banks," Sustainable Futures, vol. 5, p. 100107, 2023. DOI: 10.1016/j.sftr.2023.100107

[3] S. Huda, "AMLNet Dataset for Financial Fraud Detection," Zenodo, 2025. [Online]. Available: https://zenodo.org

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralised Data," in Proc. AISTATS, 2017, pp. 1273-1282.

[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[6] A. Hagberg, D. Schult, and P. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in Proc. 7th Python in Science Conference, 2008, pp. 11-15.

[7] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, "CopyCatch: Stopping Group Attacks by Spotting Lockstep Behaviour in Social Networks," in Proc. WWW, 2013, pp. 61-72.

[8] P. Kairouz et al., "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, vol. 14, no. 1-2, pp. 1-210, 2021.

[9] V. Upadrista, N. Bhargava, and R. Gopal, "Anti-Money Laundering (AML) Detection Platform Leveraging Federated Learning with NVIDIA FLARE," Journal of Machine Learning and Data Analytics, 2024. [Online]. Available: https://longdom.org

[10] A. O. Japinye, "Integrating Machine Learning in Anti-Money Laundering through Crypto: A Comprehensive Performance Review," European Journal of Accounting, Auditing and Finance Research, vol. 12, no. 4, pp. 54-80, 2024.

[11] J. Ma, H. Liu, M. Zhang, and Z. Liu, "VPFL: Enabling Verifiability and Privacy in Federated Learning with Zero-Knowledge Proofs," Knowledge-Based Systems, vol. 299, p. 112115, 2024. DOI: 10.1016/j.knosys.2024.112115

[12] Y. Bi, Q. Zhang, and H. Liu, "High-Throughput AML Detection with Federated Machine Learning," in Proc. ACM KDD Workshop on Fintech, 2024, pp. 56-63.

[13] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, "Deep Learning and Explainable AI in AML Detection," IEEE Access, vol. 9, pp. 45310-45330, 2021.

[14] W. W. Lo, G. K. Kulatilleke, M. Sarhan, S. Layeghy, and M. Portmann, "Inspection-L: Self-Supervised GNN Node Embeddings for Money Laundering Detection in Bitcoin," Applied Intelligence, vol. 53, no. 16, pp. 19406-19417, 2023.

[15] Z. Wang, N. Dong, J. Sun, W. Knottenbelt, and Y. Guo, "zkFL: Zero-Knowledge Proof-Based Gradient Aggregation for Federated Learning," IEEE Transactions on Big Data, 2024. arXiv: 2310.02554

[16] B. Guembe, A. Azeta, S. Misra, and R. Damasevicius, "A Federated Machine Learning Approaches For Anti-Money Laundering Detection," SSRN Electronic Journal, 2023. [Online]. Available: https://ssrn.com

[17] Effendi, F., & Chattopadhyay, A. (2024). Privacy-Preserving Graph-Based Machine Learning with Fully Homomorphic Encryption for Collaborative Anti-Money Laundering. In *arXiv (Cornell University)*. Technische Universitat Dresden. https://doi.org/10.48550/arxiv.2411.02926

[18] T. Suzumura, H. Kanezashi, and C. Choquette-Choo, "Federated graph-learning platform for anti-money laundering," in Proc. IEEE Big Data, 2019, pp. 3876-3883

[19] He, Y., & Chen, J. (2022). *AMLChain: Supporting Anti-money Laundering, Privacy-Preserving, Auditable Distributed Ledger* (pp. 50–67). Springer. https://doi.org/10.1007/978-3-030-93956-4_4

