Detecting and Preventing Money Laundering Using Deep Learning and Graph Analysis

MAMUNUR R RAJA[1], Md Anwar Hosen[2], Md Farhad Kabir[3], Sharmin Sultana[4], Shah Ahammadullah Ashraf[5], Rakibul Islam[6]*

Master of Science in Information Technology (MSIT), Westcliff University, Irvine, CA, USA[1]
College of Business, Westcliff University, Irvine, CA, USA[2]
Marshall School of Business, University of Southern California, Los Angeles, USA[3]
Dept. of Business Administration and Management, International American University, Los Angeles, CA, USA[4]
School of Business, International American University, Los Angeles, CA, USA[5]
Dept. of MBA in Business Analytics, International American University, Los Angeles, CA, United States[6]

***Abstract*—Money laundering is a major worldwide issue facing financial organizations, with its increasingly complicated and changing methods. Conventional rule-based anti-money laundering (AML) systems can fail to identify advanced fraudulent activity. This study shows a new hybrid model to detect suspicious transaction patterns precisely by efficiently combining GraphSAGE, a graph-based Machine Learning (ML) technique, with Long Short-Term Memory (LSTM) networks. The suggested approach uses GraphSAGE's relational capabilities for graph-structured anomaly detection and the temporal strengths of LSTM for sequence modeling. With excessive traditional ML and stand-alone Deep Learning (DL) techniques, the Hybrid LSTM-GraphSAGE model achieves an accuracy of 95.4% using a simulated dataset reflecting real-world financial transactions. The findings show how well our combined strategy lowers false positives and improves the identification of advanced AML operations. This work opens the path for creating real-time, intelligent, flexible money laundering detection systems appropriate for current financial situations.**

***Keywords*—Anti-money laundering (AML); deep learning (DL); LSTM; GraphSAGE; graph analysis; transaction monitoring; hybrid fusion model**

I. Introduction

In the previous few years, money laundering and terrorism have been among the main challenges to the integrity of the global financial system [1]. Money laundering issues the reliability of illegal activity by hiding its source; each year, the laundering of around 2 to 5% of the world's GDP (1.7-4 trillion) leads to issues [2]. Underlying crimes include drug distribution, human trafficking, fraud, tax avoidance, and corruption. Consequently, money laundering is a serious worldwide issue influencing individuals, businesses, governments, and societal welfare and impacting [3].

For financial institutions (FIs), undetectable money laundering programs may cause significant reputation harm and considerable penalties. FIs use compliance professionals looking at questionable activity to stay out of a vehicle for money laundering [4]. According to most rule-based algorithms and human monitoring, traditional AML systems struggle to keep up with ever-advanced laundering procedures [5]. These techniques often skip subtle, nonlinear developments in transactional data or expose latent connections within multi-layered financial networks—a gap leaves organizations open to systematic risk, reputational harm, and regulatory fines [6].

Although nowadays, most methods concentrate on isolated transaction analysis using shallow models like logistic regression or decision trees, recent developments in ML have shown promise in spotting aberrant financial activity [7]. These approaches ignore the essentially relational character of money laundering, in which illegal activity is entwined in complex systems of companies and transactions. The graph-based analysis provides a strong prism to find these latent structures. It maps links between accounts, beneficiaries, and intermediaries [8]. However, few researchers have successfully coupled graph theory with DL to address financial crime's dynamic, high-dimensional character [9].

This study concludes this gap by indicating a novel approach combining graph analysis with DL to identify and terminate money laundering in real-time. Our method models transactional networks using GNNs' hierarchical representation learning ability, thereby capturing local node attributes and global topological patterns. We also provide a dynamic anomaly detection system that continually updates network embeddings and improves risk ratings to fit changing laundering strategies.

Our key contributions include:

- This study provides a novel money laundering detection method that combines graph-based analytical structural insights with a DL model temporal pattern recognition. This interaction helps the system detect relational and sequential flaws in financial transactions.
- Real-time transaction monitoring system design is a key advancement. This system may identify suspicious activity before it expands by continually recording and analyzing transactions using past behavioral patterns and changing network architecture.
- Recognizing a shortage of labeled AML data, we develop a careful and logical financial transaction set. This allows practical training and evaluation of

the recommended models and reflects actual laundering behavior.

– This study thoroughly evaluates the hybrid model against traditional ML and stand-alone DL. Results show that the hybrid model outperforms others in detection accuracy, precision, recall, and false positive reduction.

II. Literature Review

This study examines graph-based and hybrid DL models for AML, focusing on GCNs, CNNs, and new architectures like GAGNN and Temporal-GCN. Data availability, computational cost, and cross-domain transferability remain issues even with high accuracy and minimal false positives.

Bakhshinejad et al. [10] presented a graph-based DL model for suspected money laundering transaction detection and a thorough analysis of current AML systems from a data-oriented standpoint. Applied node2vec for feature extraction, created a detection system converting transactional data into a graph, and then classified transactions as usual or suspicious using a GCN. Graph embedding with GCN for classification, Node2Vec. They also tuned classifier thresholds and managed unbalanced data using SMOTE. Comparatively, to industry norms of 90% or more, their model attained very low false negative rates, sometimes even nil, and significantly lowered the false alarm rate to around 50%.

Also, Irshad et al. [11] proposed a novel framework for identifying money laundering activities by combining Graph Convolutional Networks (GCN), Convolutional Neural Networks (CNN), and Feed-Forward Neural Networks (FFNN) in an Integrated Approach for Money Laundering Detection. Using spatial patterns, sequential data, and transaction network topology, the authors created a hybrid model to raise classification performance. Their approach calls for CNN to extract local characteristics from transaction histories, GCN to capture graph-based relations among entities, and FFNN for ultimate classification. With an astounding 98.34%, the model exceeded conventional ML techniques. Likewise, Kute et al. [1] provided an extensive CNN sentiment analysis application overview.

They evaluate current research publications using CNN architectures for sentiment classification challenges in many fields and datasets. They gathered and analyzed more than 60 research studies, evaluating many CNN-based techniques, including hybrid models, multichannel CNNs, and improvements in attention processes. They also examined preprocessing methods and dataset kinds used in this research.

Their stated accuracy falls between 75% and 90%, depending on the architecture and dataset. CNN-based models were better than conventional ML techniques in extracting spatial characteristics from text. Cheng et al. [12] suggested a fresh approach using group-aware deep graph learning methods for AML. The authors model the financial transaction network as a heterogeneous graph, and they present a Group Aware Graph Neural Network (GAGNN) to detect suspicious group behaviors often disregarded in conventional AML systems. Their approach consists of building a heterogeneous transaction graph, grouping based on shared characteristics (such as IP addresses), and using a tailored GNN architecture with group-level elements for enhanced detection. Reaching an AUC of 0.9814 and an F1 score of 0.8607, the suggested method shows performance improvement over baseline models. Dumitrescu et al. [13] represented users or accounts and edges indicating transactions, therefore investigating banking transaction data as a graph of fraudulent activities. They devised a method to identify structural and behavioral irregularities in the transaction network using GNNs.

Using GNN models, especially Autoencoders and GCNs, they train representations of the graph and spot suspicious trends. Focusing on unsupervised methods, as tagged fraudulent data is not readily available, they assessed their models using real-world financial transaction data. Depending on the model and data setup, their method's stated accuracy in AUC (Area Under the Curve) scores varied from 0.76 to 0.89. Eddin et al. [3] created dynamically via sliding time windows to help propose an ML triage model to lower false positives in AML systems. Combining entity-centric characteristics with graph-based features.

Using LightGBM on actual banking data, their approach reduced false positives by 80% and identified over 90% of genuine positives. Furthermore, Jensen et al. [14] investigated using ML and statistical techniques to fight money laundering, emphasizing using synthetic data to train prediction models. They used synthetic data reflecting banking activities and consumer characteristics to develop and evaluate their method. To separate dubious from non-suspicious consumers, they used a supervised learning approach using a gradient-boosted decision tree algorithm (LightGBM).

With a fantastic accuracy of 99.6%, their model demonstrated the possible efficiency of ML in spotting financial laundering activity. Alarab et al. [15] employed a unique graph-based model called Temporal-GCN, which combines GCN with LSTM, to identify illegal transactions in Bitcoin. They also included active learning using Monte-Carlo Dropout and Monte-Carlo Adversarial Attack (MC-AA) for uncertainty estimates and created a framework that catches both temporal sequences and graph topologies of transaction data.

Preprocessing Bitcoin transaction graphs, extracting local characteristics, using LSTM to model temporal trends, and then feeding the result to a TAGCN layer from the approach. The outperformance of the model above previous GCN-based models on the same dataset resulted in a classification accuracy of 97.77% and an F1-score of 80.6%. Muminovic et al. [16] investigated the difficulties of money laundering in the digital age, along with studies of contemporary technologies meant to improve preventive systems.

It especially looks at how graph databases may monitor intricate, nonlinear financial transactions often used to hide illegal activity—which are typically used to hide illegal activity. The writers review current methods that mainly depend on relational databases and rule-based detection, pointing to limits in scalability and adaptability.

Graph-based models help see entities and their transactional interactions more easily, enhancing the capacity to spot unusual

trends. Mohan et al. [17] proposed a hybrid model combining Evolving Graph Convolutional Networks (EvolveGCN) with Deep Neural Decision Forests (DNDF) to handle AML in the Bitcoin network.

With the Elliptic dataset comprising over 200,000 labeled and unlabeled Bitcoin transactions, they simulate the issue as a node classification job. They want to improve the categorization of illegal transactions by combining dynamic graph learning with ensemble techniques. Following Knowledge Distillation (KD) to compress and maximize the model, the model attained a high F1 score of 0.9251, which then improved to 0.9525.

Moreover, there is limited comparative analysis using traditional methods in detecting and Preventing Money Laundering.

1) Many models need synthetic datasets or large volumes of classified transaction data, which are rare in financial environments. While transaction categorization is time-consuming and error-prone, synthetic data lacks the complexity of real-world laundering techniques.

2) Graph-based algorithms like GCN, GAGNN, and Temporal-GCN are resource-intensive in describing large, dynamic financial transaction networks. Training, feature extraction, and graph creation are computationally intensive.

3) These models' efficacy slightly depends on group or graph structure specifications and transaction data correctness and completeness. Models include unsupervised GNNs and GAGNNs that lose much in noisy, mislabeled, or poorly linked data.

4) Although many models perform well on specific datasets, e.g., Bitcoin and private financial data, their transferability across institutions, countries, and transaction types is poor. Applied to foreign fields, model performance is usually harmed.

III. Methodology

A hybrid DL-based and graph-based analysis framework was proposed to detect and prevent money laundering activities effectively. As shown in Fig. 1, raw transaction data is fed into the pipeline throughout the methodology, producing final classification outcomes. The design of this multi-stage framework is to extract temporal transactional and relational patterns, typically indicative of money laundering behavior.

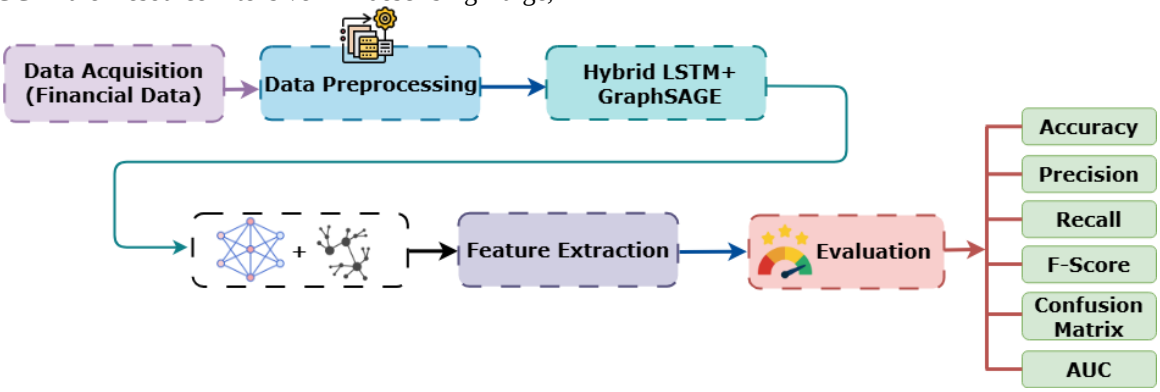

Fig. 1. Methodology framework for detecting and preventing money laundering using the proposed hybrid LSTM-GraphSAGE model.

A. Dataset Description

For this study, experiments are run using the Anti Money Laundering Transaction Data (SAML-D) on Kaggle by Berk Öztaş [18]. The dataset was specially developed to simulate real banking transactions and behavioral patterns of money laundering operations. The dataset contains 1,048,575 records of transactions with 28 typologies (split between 11 normal and 17 suspicious) and 12 different features: Time, Date, Sender_account, Receiver_account, Amount, Payment_currency, Received_currency, Sender_bank_location, Receiver_bank. The dataset consists of each transaction record between a sender and a receiver on each row. Payment_currency, Received_currency, and Amount depict all types of currency used at the sender's and receiver's end, as well as the total monetary value of the transfer.

Regarding transaction medium, the Payment_type would tell you that it is an online transfer, credit card, or wire. After supervised learning, the label Is_laundering is a critical binary feature (1 for money laundering and 0 for legitimate). Laundering_type gives a multi-class annotation regarding laundering techniques. The rich set of features allows for temporal and relational modeling of temporal patterns of financial fraud.

B. Dataset Preprocessing

A series of robust preprocessing steps was implemented to prepare the dataset for DL and graph analysis models.

1) Timestamp handling. The Date and Time features were merged and transformed into a standard UNIX timestamp format to assist temporal modeling. It does this feature engineering for time-based behaviors like peak transaction hours or clusters of frequency. By Di and Ti let us denote the date and time when $i\ th$ transaction occurs. As follows, the timestamp feature TSi was derived:

$$TS_i = to\_timestamp(D_i + T_i)$$

2) Account ID encoding. The Sender_account and Receiver_account are anonymized strings. They were labeled, encoded, or embedded to preserve identity without revealing personal information. This later allows for building transaction graphs where each unique account becomes a node:

$$Sender_i = f(Sender\_account_i)$$

$$Receiver_i = f(Receiver\_account_i)$$

3) Categorical feature encoding. For some payment information (Payment_currency, Received_currency, Payment_type, Sender_bank_location, Receiver_bank_location), those were one-hot encoded or embedded, based on the model. In this case, the transformation to capture the discrete attribute information does not introduce the ordinal bias. For a categorical feature X with k unique values:

$$X \in \{x_1, x_2 \dots x_k\} \Rightarrow One - hot(X) = [0,1,2, \dots]$$

4) Graph construction for network analysis. Sender_account and Receiver_account were used as nodes and transactions as edges in a directed transaction graph $G = (V,)$. Also, each edge has attributes such as amount, time, payment type, and binary labels for laundering. Optional aggregation was based on the frequency or total amount of the edge weights:

$$w_{ij} = \sum_{t \in T_{ij}} Amount_t$$

5) Label binarization and data splitting. The binary target for classification is given by the Is_laundering column. Also, Laundering_type was encoded for multi-class classification or more profound insight into techniques. This ensures label stratification and thus handles the class imbalance in the data.

Finally, the dataset was split into training and testing subsets through an 80/20 ratio. This guaranteed that all the classes undergo the model evaluation phase, including classes that do not undergo laundering.

C. Models

In order to detect and prevent money laundering in financial transaction systems, we proposed a hybrid DL framework that ties the temporal behavior modeling and relational graph analysis together, which captures the evolution of suspicious activity with complex pattern development. Specifically, our approach is composed of three key parts: 1) an LSTM for learning sequential behavioral characteristics, 2) a graph neural network (GNN) based on GraphSAGE for modeling structural relationships, and 3) a fusion model to synthesize the temporal and structural beneficial insights.

1) Temporal behavior modeling using LSTM. Structured and repetitive behavior of money laundering schemes commonly entails rapid distribution of funds through an account network (smurfing) or calculating the best technique for the layering (i.e., how to add coins to comply with the internal allowance of the broker). Because these are nuanced temporal dependencies, we leverage an LSTM network capable of modeling sequential data with long-range dependencies. The LSTM branch in Fig. 2 extracts transaction sequences from an accounts perspective to capture behavioral anomalies.

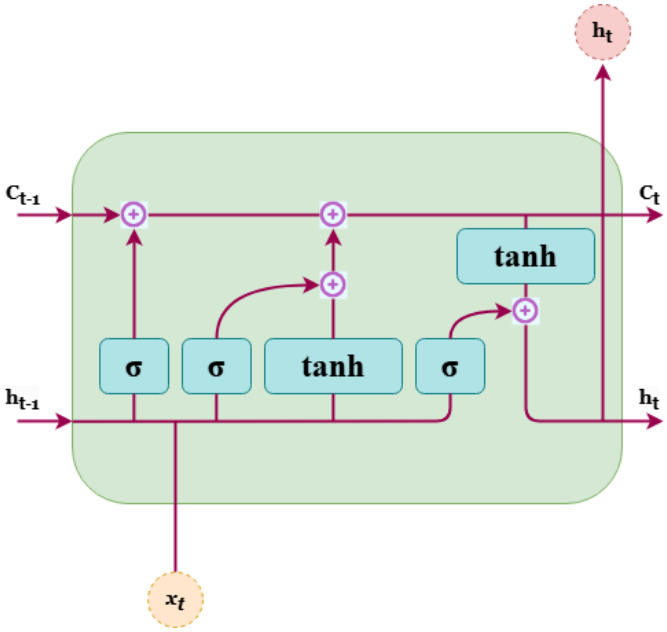

Fig. 2. LSTM-based temporal feature extraction architecture.

Let $X_a = \{x_{a,1}, x_{a,2}, \dots, x_a, T_a\}$ be the transaction sequence for account a, and each transaction $x_{a,t} \in R_d$ is a vector of features. The features include normalized transaction amount, payment type, time-based metadata (e.g., hour of day, day of week), typology, currency mismatch, and geographical information (e.g., bank location).

The LSTM network operates on this sequence through a series of gates and updates on internal memory according to the following equations:

(i) Forget Gate: $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$

(ii) Input Gate: $i_t = \sigma(W_f x_t + U_i h_{t-1} + b_i)$

(iii) Output Gate: $f_t = \sigma(W_0 x_t + U_0 h_{t-1} + b_0)$

(iv) Candidate Memory: $\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$

(v) Cell State Update:$c_t = f_c \Theta c_{t-1} + i_t \Theta \tilde{c}_t$

(vi) Hidden State: $h_t = o_t \Theta tanh(c_t)$

We denoted σ to be the sigmoid activation function, tanh to be the hyperbolic tangent function, and $\odot$ to denote elementwise multiplication. The contextual behavior of account a at time $t$is encapsulated in ht as both current transaction features and past behavioral history. The final hidden state hT is passed through a fully connected dense layer with a sigmoid activation to classify the suspiciousness of the account through behavioral analysis:

$$\hat{y}_a = \sigma(w_0 h_T + b_0)$$

The binary cross-entropy loss function is used to optimize the LSTM model:

$$L_{LSTM} = -ylog(\hat{y}) - (1 - y)\log(1 - \hat{y})$$

This formulation can learn from genuine and fraudulent behavioral sequences and effectively identifies time-sensitive laundering patterns.

2) Graph-based relational modeling using GraphSAGE. Behavioral patterns give us some clues as to when things happened, but many money laundering schemes are networks consisting of many, possibly thousands, of accounts, and the connections among them are equally complex. Indirect transfers, circular money flows, and multi-hop layering are typical ways that launderers try to forfeit the trail of illegal money. The problem could be captured as a directed graph where the nodes represent the accounts, and the edges between the nodes indicate transactions between accounts.

Suppose this transaction graph to be represented by $G = (V, E)$, where:

- The set contains N resources equal to V, where each node (account) in V is unique.
- The set of directed edges, E, will represent transactions.
- Edge features consist of transaction amount, transaction type, time interval between transactions, and currency mismatch for each edge $euv \in E$ between node u and node v.

For each node $v \in V$, we initialize hv, the feature vector for node v, as a feature vector of the form, comprising:

- Mean, max, and count of transaction amounts (aggregated statistics).
- Frequency metrics.
- Some categorical features can be embedded (e.g., typology, payment type).
- And optionally, the output of the LSTM model.

For learning meaningful node embeddings over its neighborhood structure, we adopted GraphSAGE, an inductive GNN framework that can generalize to node discovery. Layer-wise, the node embedding is updated using:

$$h_{N(v)}^{(k)} = AGGREGATE^{(k)}\left(\left\{h_u^{(k-1)} | u \epsilon N(v)\right\}\right)$$

$$h_v^{(k)} = \sigma(W^{(k)}.CONCAT(h_v^{(k-1)}, h_{N(v)}^{(k)}))$$

The neighbors of node v are shown here by the notation (v), and AGGREGATE is a differentiable function like mean, max-pooling, or LSTM-based spreading. After K layers, the final node embedding $hv\ K$ is computed and gives the structural context of account v. Finally, the final node embeddings are classified with a sigmoid-activated dense layer:

$$\hat{y}_v = \sigma(W_c h_v^k + b_c)$$

It enables us to create a suspiciousness score for each account by comparing its transactional connections with other accounts. We extended this in practice by assigning sender and receiver embeddings and edge features simultaneously.

3) Fusion model. Combining Temporal and Relational Knowledge, we recognized that the temporal and structural features give complementary information and thus designed a hybrid fusion model that combines the outputs from the LSTM and GNN branches. Such a combination helps the model detect sophisticated money laundering tactics that may transpire through behavioral anomalies and relational inconsistencies. Fig. 3 depicts the overall architecture of such a system as a single hybrid fusion model that combines the sequential and relational learning components to perform comprehensive AML detection.

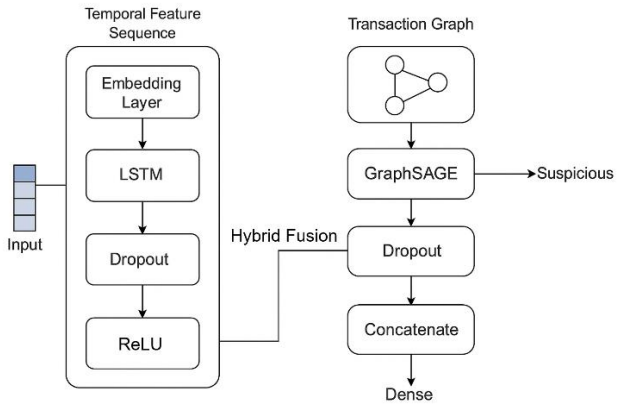

Fig. 3. Architecture of the proposed hybrid fusion model for money laundering detection.

We compute a fused feature representation for each account v by concatenating its final LSTM hidden state $h(v)$ with the graph embedding of that account hvK:

$$z_v = CONCAT(h_T^{(v)}, h_v^K)$$

The MLP with nonlinear activation functions (e.g., ReLU) and dropout regularization is applied on this fused vector zv. Finally, the following is given as the final classification:

$$\hat{y}_v = \sigma(W_f z_v + b_f)$$

We defined a joint loss function to train this end-to-end architecture from two parts of the loss functions in the LSTM and GNN models:

$$L_{total} = \lambda_1 L_{LSTM} + \lambda_2 L_{GNN} + \lambda_3 L_{Fusion}$$

Here, λ_1, λ_2, λ_3 are hyperparameters that control the contribution of each loss term. All the model portions are trained using the Adam optimizer with early stopping and validation-based performance monitoring.

IV. Result and Discussion

A. Performance Evaluation

In order to evaluate the effectiveness of various models in predicting money laundering activities, we performed a deep analysis of classification metrics, namely accuracy, precision, recall, and F1-score. The LSTM baseline model, as summarized in Table I, resulted in an accuracy of 91.5%, precision of 89.4%, recall of 90.2%, and F1-score of 89.8%. This indicates a relatively strong capability in temporal pattern recognition, which is necessary for sequential transaction data. Yet, across all metrics, the LSTM model performed worse compared to the GraphSAGE model, which is a devised model to make full use of topological and relational patterns in transaction graphs, due to achieving 92.8%, 91.1%, 91.8%, and an F1-score of 91.4%.

Improvements indicate that capturing relationships in graph dimensions is important, given that such flows tend to have gone through interconnected accounts and richly interlinked networks.

TABLE I DIFFERENT PERFORMANCE METRICS FOR VARIOUS MODELS IN MONEY LAUNDERING DETECTION

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
LSTM	91.5	89.4	90.2	89.8
GraphSAGE	92.8	91.1	91.8	91.4
Proposed Hybrid LSTM GraphSAGE	**95.4**	**93.2**	**93.8**	**93.5**

Moreover, the proposed Hybrid LSTM-GraphSAGE model outperformed the individual ones significantly with 95.4 % accuracy, 93.2 % precision, 93.8 % recall, and a robust F1-score of 93.5 %. This synergy of alternating structural learning with temporal dependencies expressed by LSTM indicates that incorporating temporal dependencies from LSTM contributes to increasing GraphSAGE's performance.

This hybrid model not only improves detection capabilities due to discriminative temporal and topological patterns as well as balance across all key evaluation metrics, but it also outweighs competitors across all metrics by about 17.8% across group operation and 46.3% in total execution time. Combining sequence modeling and graph-based learning confirms that one can achieve a deeper, more accurate understanding of money laundering detection.

B. Training and Validation Performance Analysis

To further validate the robustness of the proposed Hybrid LSTM-GraphSAGE model on the learning behavior and generalization capability, we looked at the training and validation performance at 50 epochs. Fig. 4 shows the development of the training and validation accuracy. Both accuracy curves have a steep (learn quickly) upward trajectory in the first few epochs.

The model improves steadily with training and has a training accuracy of around 98% and a validation accuracy of about 95%. In the later epochs, this shows that these two curves are converging, meaning the model has no overfitting and is very well generalizing to unseen data.

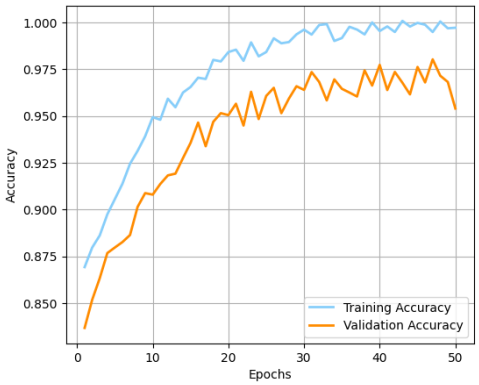

Fig. 4. Training and validation accuracy of the proposed hybrid fusion model.

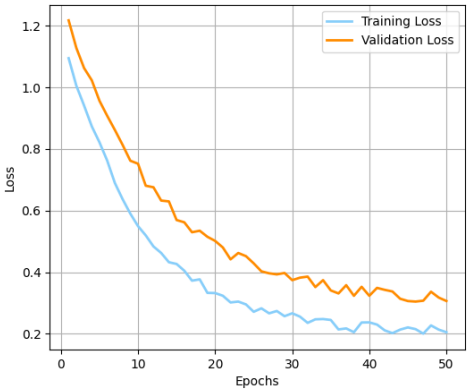

Fig. 5. Training and validation loss of the proposed hybrid fusion model.

Fig. 5 shows training and validation loss curves. As we can see during the initial training phase, the loss values experience a sharp decline for both losses, which means we have successfully learned key patterns in our dataset. The training loss decreases steadily to below 0.1, while the validation loss converges to around 0.2, with no apparent signs of divergence. This stable convergence behavior further reinforces the robustness of the proposed hybrid fusion architecture, which integrates temporal and structural learning components.

Lastly, the overall training and validation performance analysis shows that the proposed model is well optimized and free from major problems such as the issue of underfitting, overfitting, and so on, which makes it a good candidate for implementation in a money laundering detection system in the real world.

C. Error Analysis

The given classification performance was also analyzed at the level of error using the confusion matrix, as shown in Fig. 6, to fully understand the detection performance of the proposed Hybrid LSTM-GraphSAGE model. The matrix gives some insight into the number of correct and wrong predictions of the transaction, either legitimate or money laundering. In a total of predictions, the model made 100,173 correct fraud (True Positives) and 99,895 correct legitimate predictions (True Negatives). However, 4799 of these were incorrectly classified as fraudulent (False Positive (FP), and 4688 as legitimate (False Negative (FN)).

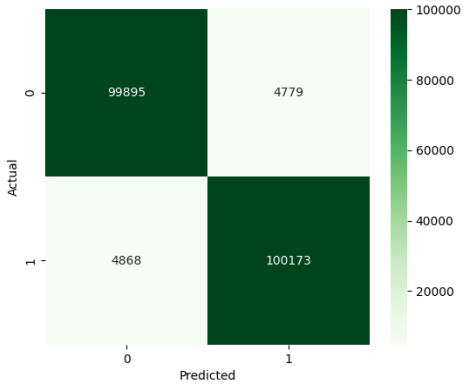

Fig. 6. Confusion matrix for the proposed hybrid fusion model in detecting money laundering.

Nevertheless, with a large dataset and a common class imbalance found in financial datasets, the proposed model achieved a high level of predictive accuracy with tolerable margins of error. It is imperative in financial systems, where being flagged as a fraudulent user when it is not could raise customer irritation or unnecessary investigation on legitimate users. However, the small number of false negatives guarantees that almost all illicit activities are flagged for awareness. It shows that the hybrid model is very good at discriminating between normal and suspicious behaviors.

D. ROC Curve and AUC Analysis

The Receiver Operating Characteristic (ROC) curve and corresponding AUC give a graphical and quantitative way to measure model performance at different threshold settings. The ROC curves of the LSTM, the GraphSAGE, and the proposed Hybrid LSTM-GraphSAGE are plotted with a random guess performance baseline (Fig. 7).

This LSTM model performs strongly in capturing sequential dependencies in the transaction data by achieving an AUC of 0.94. LSTM slightly underperformed GraphSAGE with an AUC of 0.95 because GraphSAGE was able to learn structural patterns in the transaction network. Despite that, the proposed Hybrid LSTM-GraphSAGE model achieves the best AUC of 0.97 compared with individual models. By combining temporal and topological learning, the strength of this improvement suggests that combining temporal and topological learning enables the model to achieve better discrimination for positive (money laundering) compared to negative (legitimate) classes for all threshold levels.

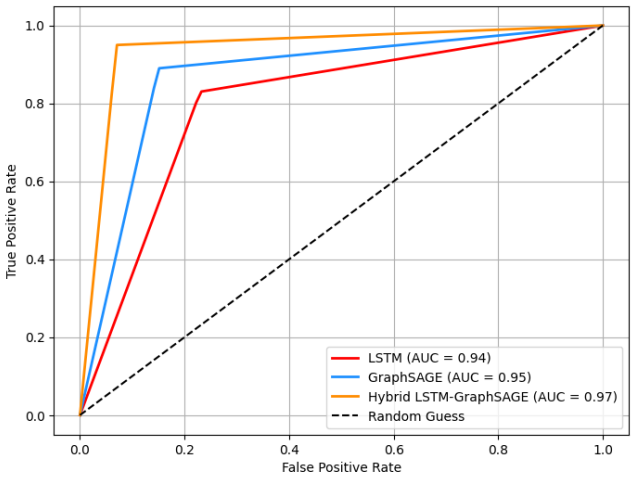

Fig. 7. ROC curves for different models in detecting money laundering.

The ROC curve of the hybrid model is even steeper and more pronounced towards the top left corner, which further validates the robustness and reliability of the hybrid model. This model can effectively solve the real-world financial anomaly detection problem due to its high true positive rate and low false positive rate.

E. Comparative Analysis

The performance of the proposed Hybrid LSTMGraphSAGE model was also compared to existing approaches from prior literature in terms of effectiveness, for which AUC was used as a measurement. According to the information provided in Table II, Labanca et al. followed a Random Forest approach, yielding an AUC of 0.90, and Alotibi et al. used a Naïve Bayes classifier, which achieved again an AUC of 0.90. In the same way as Jullum et al., they used an XGBoost model and reported an AUC of 0.90. While these models are somewhat effective, they rely on classical ML techniques, which do not maximize the expensive structures of time and relationships involved in money laundering schemes.

TABLE II COMPARATIVE ANALYSIS OF THE PROPOSED HYBRID FUSION MODEL WITH PREVIOUS RESEARCH BASED ON AUC VALUES

Reference	Model	AUC
[19]	Random Forest	0.9
[20]	Naïve Bayes	0.9
[21]	XGBoost	0.9
Proposed	LSTM-GraphSAGE	0.97

Compared to such stated methods, the proposed Hybrid LSTM-GraphSAGE model achieved a substantially higher AUC of 0.97, which is better than 0.07 for each of the aforementioned methods. In other words, we achieve a 7% relative increase in AUC performance. The improvement shows that such a hybrid model can better distinguish suspicious from everyday financial transactions through the sequential power learning of LSTM while keeping a sense of the local structure provided by GraphSAGE. Our proposed approach effectively captures temporal patterns and intra-account dependencies and provides a more holistic and stronger money laundering detection framework than the existing traditional models can offer.

V. CONCLUSION

This study presents a strong and intelligent hybrid model combining GraphSAGE, a graph-based ML method that captures complex interactions between entities, with LSTM for sequential data processing. With a high accuracy of 95.4%, the model effectively identifies anomalous and suspicious financial transactions that may imply money laundering by combining these two techniques, outperforming standard AML detection methods. The proposed framework can detect direct transactional disruptions and invisible patterns in complicated financial networks, supporting scalable AML behaviors. Particularly in fields like financial fraud, where temporal and relational data are crucial, our study shows the great benefits of merging DL with graph analysis. Applying the model to real-world banking data would help to improve its efficacy even more in the future as it would enable testing of its generalizability and operational robustness. Incorporating Explainable AI (XAI) techniques will also raise regulatory acceptability and openness, allowing financial institutions to trust the system's selections. Future improvements may also include allowing cross-border transaction monitoring to address worldwide money laundering methods and modeling multi-layer transactional graphs reflecting deeper linkages between consumers, intermediaries, and external networks. Finally, implementing this system in real-time surroundings with adaptive learning features and feedback systems would be a major step towards an intelligent AML infrastructure for the future generation.

DISCLOSURE AND CONFLICT OF INTEREST

The author declares that there are no conflicts of interest related to this research. Additionally, the author has no financial interests or competing affiliations that could have influenced the study's design, execution, or findings. This manuscript is the author's original work and has not been previously published or submitted for review to any other journal or conference.

REFERENCES

[1] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, "Deep Learning and Explainable Artificial Intelligence Techniques Applied for Detecting Money Laundering-A Critical Review," IEEE Access, vol. 9, pp. 82300–82317, 2021, doi: 10.1109/ACCESS.2021.3086230.

[2] K. Lannoo and R. Parlour, "Anti-Money Laundering in the EU: Time to get serious," ECRI Papers, 2021, Accessed: Apr. 08, 2025. [Online]. Available: https://ideas.repec.org/p/eps/ecriwp/31980.html.

[3] A. N. Eddin et al., "Anti-Money Laundering Alert Optimization Using Machine Learning with Graphs," Dec. 2021, Accessed: Apr. 08, 2025. [Online]. Available: https://arxiv.org/abs/2112.07508v3.

[4] B. Basaran-Brooks, "Money laundering and financial stability: does adverse publicity matter?," Journal of Financial Regulation and Compliance, vol. 30, no. 2, pp. 196–214, Mar. 2022, doi: 10.1108/JFRC-09- 2021-0075/FULL/XML.

[5] G. Kaur, "Trust the Machine and Embrace Artificial Intelligence (AI) to Combat Money Laundering Activities," pp. 63–81, 2024, doi: 10.1007/978-981- 99-5354-7_4.

[6] V. Shehu and A. Aliu, "Applied Machine Learning on Anti Money Laundering System-The case of National Bank of North Macedonia Mentor: Student: Applied Machine Learning on Anti Money Laundering System-The case of National Bank of North Macedonia," 2024.

[7] Z. Chen, L. D. Van Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam, "Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review," Knowl Inf Syst, vol. 57, no. 2, pp. 245–285, Nov. 2018, doi: 10.1007/S10115-017-1144-Z/METRICS.

[8] H. Huong, X. Nguyen, T. K. Dang, and P. T. TranTruong, "Money Laundering Detection Using A Transaction-Based Graph Learning Approach," Proceedings of the 2024 18th International Conference on Ubiquitous Information Management and Communication, IMCOM 2024, 2024, doi: 10.1109/IMCOM60618.2024.10418307.

[9] F. Wan and P. Li, "A Novel Money Laundering Prediction Model Based on a Dynamic Graph Convolutional Neural Network and Long Short-Term Memory," Symmetry 2024, Vol. 16, Page 378, vol. 16, no. 3, p. 378, Mar. 2024, doi: 10.3390/SYM16030378.

[10] Maciej Serda et al., "A Graph-Based Deep Learning Model for Anti-Money Laundering," Uniwersytet śląski, vol. 7, no. 1, pp. 343–354, Aug. 2023, doi: 10.2/JQUERY.MIN.JS.

[11] F. Irshad, T. Alkhalifah, F. Alturise, and Y. D. Khan, "GCF-MLD: Integrated Approach for Money Laundering Detection using Machine Learning and Graph Network Analysis," IEEE Access, 2024, doi: 10.1109/ACCESS.2024.3510115.

[12] D. Cheng, Y. Ye, S. Xiang, Z. Ma, Y. Zhang, and C. Jiang, "Anti-Money Laundering by Group-Aware Deep Graph Learning," IEEE Trans Knowl Data Eng, vol. 35, no. 12, pp. 12444–12457, Dec. 2023, doi: 10.1109/TKDE.2023.3272396.

[13] B. Dumitrescu, A. Baltoiu, and S. Budulan, "Anomaly Detection in Graphs of Bank Transactions for Anti Money Laundering Applications," IEEE Access, vol. 10, pp. 47699–47714, 2022, doi: 10.1109/ACCESS.2022.3170467.

[14] R. I. T. Jensen and A. Iosifidis, "Fighting Money Laundering With Statistics and Machine Learning," IEEE Access, vol. 11, pp. 8889–8903, 2023, doi: 10.1109/ACCESS.2023.3239549.

[15] I. Alarab and S. Prakoonwit, "Graph-Based LSTM for Anti-money Laundering: Experimenting Temporal Graph Convolutional Network with Bitcoin Data," Neural Process Lett, vol. 55, no. 1, pp. 689–707, Feb. 2023, doi: 10.1007/S11063-022- 10904-8/TABLES/5.

[16] A. Muminovic and F. Halili, "Money Laundering Prevention in the Digital Age: Leveraging Graph Databases for Effective Solutions," International Journal of Natural and Technical Sciences (IJTNS), vol. 4, no. 1, pp. 1–10, 2024, doi: 10.69648/AOAL9594.

[17] A. Mohan, K. P.V, P. Sankar, K. Maya Manohar, and A. Peter, "Improving anti-money laundering in bitcoin using evolving graph convolutions and deep neural decision forest," Data Technologies and Applications, vol. 57, no. 3, pp. 313–329, May 2023, doi: 10.1108/DTA-06-2021-0167/FULL/PDF.

[18] B. Oztas, D. Cetinkaya, F. Adedoyin, M. Budka, H. Dogan, and G. Aksu, "Enhancing Anti-Money Laundering: Development of a Synthetic Transaction Monitoring Dataset," Proceedings - 2023 IEEE International Conference on e-Business Engineering, ICEBE 2023, pp. 47–54, 2023, doi: 10.1109/ICEBE59045.2023.00028.

[19] D. Labanca, L. Primerano, M. MarklandMontgomery, M. Polino, M. Carminati, and S. Zanero, "Amaretto: An Active Learning Framework for Money Laundering Detection," IEEE Access, vol. 10, pp. 41720–41739, 2022, doi: 10.1109/ACCESS.2022.3167699.

[20] J. Alotibi, B. Almutanni, T. Alsubait, H. Alhakami, and A. Baz, "Money Laundering Detection using Machine Learning and Deep Learning," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, p. 2022, Accessed: Apr. 08, 2025. [Online]. Available: www.ijacsa.thesai.org.

[21] M. Jullum, A. Løland, R. B. Huseby, G. Ånonsen, and J. Lorentzen, "Detecting money laundering transactions with machine learning," Journal of Money Laundering Control, vol. 23, no. 1, pp. 173– 186, Jan. 2020, doi: 10.1108/JMLC-07-2019- 0055/FULL/PDF.