УДК 004.056.5 **DOI:10.30837/rt.2026.3.226.06**

V. PROSOLOV, G. KHALIMOV, Doctor of Technical Sciences

ASSESSING OF BLOCKCHAIN TRANSACTIONS USING GATV2 AND GRAPH-SMOTE

Introduction

Modern blockchain systems have become an important element of the digital financial infrastructure; however, their openness, global nature, and pseudo-anonymity create favorable conditions for transactional threats: money laundering, financing of shadow markets, accumulation of payments following ransomware attacks, and the use of mixers and multi-step obfuscation schemes. Unlike traditional centralized information systems, a blockchain ledger has a natural graph structure in which not only the behavior of an individual address or transaction is significant, but also the context of its connections with neighboring nodes.

In the tasks of detecting financial fraud and investigating cybercrimes, the architecture, topological structure, and mathematical quality of the input data play a key role. It is precisely the characteristics of the graph set that determine the ability of neural network models to distinguish legitimate transactions from shadow ones. The specific nature of Bitcoin on-chain data presents three fundamental challenges for machine learning: extreme class imbalance, high feature dimensionality, and widespread unlabelled nodes.

Class imbalance arises because the proportion of confirmed illicit transactions is negligible compared to the volume of legitimate daily transactions. For the Elliptic dataset, the minority class accounts for about 2 % of labelled nodes, while a significant portion of transactions remains in the Unknown class. Under such conditions, standard optimization for overall accuracy can lead to a trivial solution: the model classifies most objects as legitimate, formally demonstrating high accuracy, but practically loses the ability to detect dangerous cases.

The aim of this paper is to improve the method for assessing transaction threats in blockchain systems based on latent oversampling of graph embeddings formed by the GATv2 architecture, as well as to analyse its effectiveness compared to baseline neural network models.

Related works

Current research in the field of blockchain system security increasingly views the transaction ledger as a complexly structured graph rather than a collection of independent tabular records. This is because the signs of financial cybercrimes often manifest not at the level of a single transaction, but in the relationships between addresses, temporal asset movement paths, local subgraph density, and recurring obfuscation patterns. The primary data source for such studies is the Elliptic Data Set [1], which contains Bitcoin network transactions, time slices, and an extended feature space for classifying legitimate and illicit transactions. It is precisely the presence of a graph structure and a significant proportion of unlabelled nodes that makes this dataset representative for testing risk-scoring methods under conditions of pseudonymity.

The general direction of development of methods for detecting anomalies in graph data is systematized in the work by Ma X and co-authors [2], where it is shown that deep learning on graphs allows combining the attribute characteristics of nodes with the topological context of their surroundings. For cybersecurity tasks, this is of fundamental importance, since attacks can masquerade as normal operations based on local parameters but leave a persistent trace in the structure of connections. At the same time, studies [3] on sequential and federated anomaly detection emphasize the importance of correctly modelling rare events and reducing the number of false negatives. This directly corresponds to AML/CFT tasks, where missing a risky transaction is more critical than a single false positive.

A separate group of works consists of studies focused on detecting illegal or suspicious Bitcoin transactions using machine learning methods. In particular, the work by Shaizat and Mussiraliyeva [4] emphasizes improving the quality of identifying illicit Bitcoin transactions through the selection

of informative features. This approach confirms the importance of a high-quality feature space, but does not fully eliminate the problem of losing topological context in non-graph models. In the work by Aladhadh et al. [5], the BChainGuard framework is proposed for detecting cyber threats in the blockchain using machine learning, demonstrating the practical feasibility of automated monitoring, but leaving open the question of handling extremely imbalanced graph samples.

The problem of explainability and interpretation of graph model decisions is also being actively researched. In the work by Luša, Pintar, and Vranić [6], an approach to explainable edge-aware graph neural networks for intrusion detection tasks is proposed. This is particularly important for blockchain analytics, as the model must not only flag a transaction as risky but also provide grounds for further audit or digital forensic investigation. Thus, the use of attention mechanisms in graph neural networks is promising not only in terms of classification accuracy but also in terms of generating interpretable risk scoring.

Data balancing methods represent another important area of research. Ashfaq et al. [7] explore the combination of machine learning and blockchain technologies to develop an effective fraud detection mechanism. However, standard approaches to oversampling, particularly SMOTE, were developed primarily for tabular data and do not account for the presence of edges between nodes. Therefore, in the case of transaction graphs, simply generating synthetic vectors can lead to the appearance of isolated objects that do not have a correct place in the graph structure.

Graph Attention Networks (GATv2) with multi-head attention mechanisms architecture

Transactional threats, particularly ransomware payments and multi-stage money laundering, often lack a consistent signature at the level of an individual record but leave a trace in the graph structure: repeated transfer routes, concentration of flows, anomalous subgraph density, and characteristic time intervals between operations.

In the proposed methodology, the blockchain ledger is represented as a directed attributed graph. This representation allows for the preservation not only of local transaction attributes but also of the relationships between them, which is essential for detecting multi-step obfuscation schemes.

Mathematically, the transaction graph is formalized as follows:

$$G = (V, E, X), \quad X \in \mathbb{R}^{|V| \times d}, \tag{1}$$

where V is the set of graph nodes (transactions or addresses), E is the set of directed edges describing asset flows, X is the matrix of node features, and d is the dimension of the feature space.

The first-generation Graph Attention Networks (GAT) architecture introduced anisotropic aggregation, where each connection receives its own weight (attention coefficient)α_{ij} [8]. However, the classical GAT uses a static scoring function. Mathematically, this is expressed as follows: first, the node features are linearly transformedWh_i, then concatenated and multiplied by a common weight vector a. Due to the linearity of the operation$a^{\top}[Wh_i \,||\, Wh_j]$, the ranking of neighbors j for a given node i de facto depends only on the neighbors themselves and does not adapt to the specifics of the query from the target node i.

To overcome this fundamental limitation, the GATv2 (dynamic attention) architecture [9] is implemented in the proposed method. In GATv2, the order of operations has been changed: the nonlinear LeakyReLU function is applied to the multiplication by the attention vector **a**. This mathematically transforms the attention layer into a universal approximator on graphs [3, 6].

The attention coefficiente_{ij} (before normalization) in GATv2 is calculated using the formula:

$$e_{ij} = \mathrm{a}^{T}\,\mathrm{LeakyReLU}\big(\mathrm{W}\big[h_i \| h_j \| e_{ij}^{attr}\big]\big), \tag{2}$$

where h_i and h_j are the hidden node vectors,e_{ij}^{attr} is the edge attributes, and W and a are the training parameters of the attention layer.

Normalization of attention weights is performed using the softmax function around the target node:

$$\alpha_{ij} = \text{Softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(e_{ik})}, \quad (3)$$

where N(i) is the set of neighbors of node **i**. This allows the model to focus on the connections that form suspicious transaction patterns.

To stabilize the training process and improve the model's expressiveness (expressive power), multi-head attention is applied. Instead of a single calculation, the model uses K independent attention mechanisms (heads) in parallel, each of which has its own weight matrices$W^{(k)}$ and$a^{(k)}$. According to the configuration of the developed software package (AdvancedGNN class, Appendix B), the outputs of all K heads for node i are concatenated in the hidden layers, forming an extended latent vector:

$$h_i' = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{(k)} W^{(k)} h_j \right). \quad (4)$$

The practical significance of the multi-head architecture in the context of blockchain threats lies in the semantic specialization of the space. Each of the k heads is tuned to recognize a specific pattern. Empirical studies conducted as part of this work confirmed that the optimal configuration for Bitcoin transaction graphs is K=4 attention heads (parameter heads=4).

The architecture of the developed hybrid model is constructed as a deep multi-layer pipeline that sequentially performs feature extraction, spatial aggregation, and nonlinear classification. A detailed structural diagram of the model is shown in Fig. 1

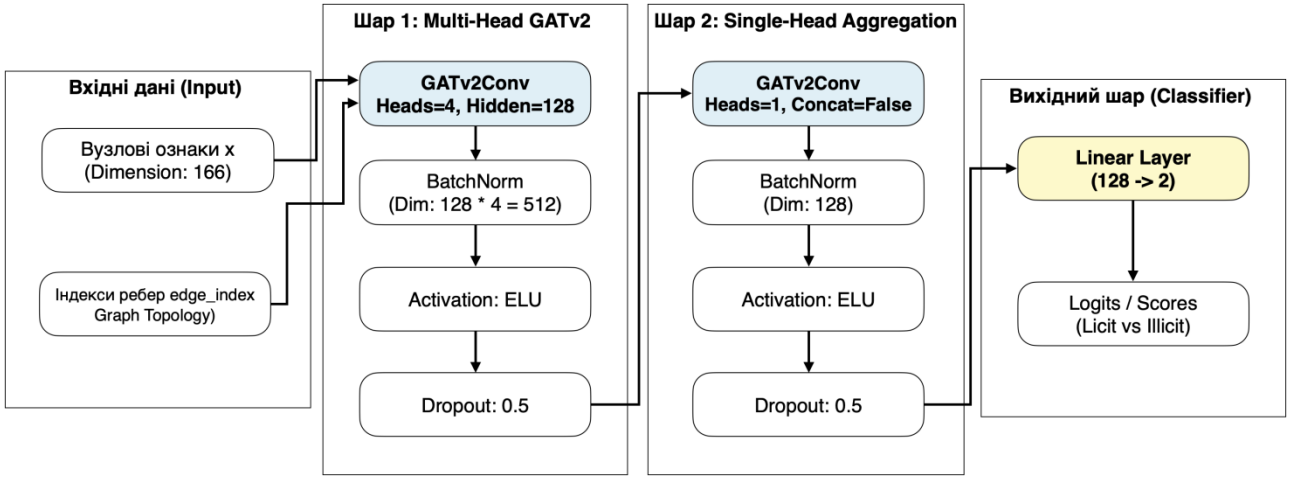


Fig. 1. Architecture of the GATv2 multi-layer graph neural network with multi-head attention

As shown in Fig. 1, the inference process begins with the input of a normalized matrix of node features ($X \in R^{N \times 166}$) and a sparse graph structure description (edge_index). The first spatial block, GATv2Conv, aggregates features of neighboring nodes using four attention heads, which allows for the formation of an expanded latent representation of dimension$128 \times 4 = 512$. This is followed by batch normalization, ELU nonlinear activation, and dropout regularization set to 0.5, which reduces the risk of overfitting on the majority class of legitimate transactions.

The second GATv2Conv block performs deep aggregation within a 2-hop neighborhood and generates a generalized latent representation of dimension 128. It is at this level that the model captures the high-level semantics of multi-step transactional patterns, including money laundering chains, mixing, and star-shaped asset consolidation. The final stage is a fully connected classification layer, which transforms the node's latent vector into a logit space for binary classification: 0 – legitimate transaction, 1 – illicit transaction or attack.

The main hyperparameters of the model are shown in Table 1. Their selection aims to ensure a balance between classification accuracy, robustness to class imbalance, and computational efficiency of the model.

Table 1

Hyperparameter configuration of the GATv2 hybrid model

Parameter (Hyperparameter)	Value	Mathematical and logical justification
Input Features (F)	166	Number of local and structural features of a node (Elliptic standard)
Hidden Channels	128	Dimensionality of the hidden latent space. Ensures a balance between inference speed and pattern extraction depth
Attention Heads (K)	4	Number of parallel attention mechanisms for capturing different types of threats (peel chains, mixers)
Dropout Rate	0.5	Regularization level; blocks 50% of connections during training to improve robustness
Learning Rate (LR)	0.005	Initial optimizer step. Ensures smooth gradient descent
Weight Decay (L2)	10^{-3}	L2 regularization coefficient; penalizes the model for overly large weights, preventing overfitting

The model is trained using the AdamW optimizer with parameters lr=0.005 and weight_decay=103

This approach allows for the correct application of weight regularization and improves the model's generalization ability, which is particularly important when using synthetic examples generated by the Graph-SMOTE mechanism.

To compensate for extreme class imbalance, a weighted cross-entropy function is used:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \log\left(\frac{\exp(z_{i,y_i})}{\sum_c \exp(z_{i,c})}\right), \quad (5)$$

where `$z_i$` is the output logit for node `$i$`, and `$y_i$` is its ground truth label. An increased weight is assigned to the minority class "Illicit," causing the gradients to heavily penalize the neural network for false negatives (missed cyberattacks) [4].

Topological oversampling techniques application(Graph-SMOTE)

The key challenges when training models to detect transactional threats in blockchain systems is the significant imbalance between the number of legitimate and illicit transactions. In real on-chain datasets, the number of confirmed malicious or suspicious nodes is significantly smaller compared to the majority class of legitimate transactions. Under such conditions, a neural network model tends to shift the decision boundary toward the majority class, leading to an increase in the number of false negatives, i.e., missed attacks.

The traditional Synthetic Minority Over-sampling Technique (SMOTE) algorithm solves the data imbalance problem by generating synthetic examples of the minority class based on linear interpolation between existing minority samples and their nearest neighbours in the multidimensional feature space [7]. However, the direct application of classical SMOTE to transaction graphs$G = (V, E, X)$ is methodologically limited. Synthetic vectors generated in this way do not have natural incoming and outgoing edges and, therefore, are not fully integrated into the graph structure. As a result, the message passing mechanism in a graph neural network cannot correctly account for such isolated nodes during the aggregation of neighborhood features [10].

In view of this, the proposed approach employs the topology-oriented oversampling method Graph-SMOTE, which shifts the generation of synthetic examples from the space of original features to the latent space of graph embeddings. This approach allows for the formation of new representations of the minority class after the initial graph aggregation, i.e., taking into account the local structural context of the nodes.

The topological oversampling process consists of three main stages. In the first stage, latent representations of nodes are extracted. The initial feature matrix$X$ and the graph structure, defined

by the set of edges$E$, are fed into the first spatial layer of GATv2Conv. As a result, a set of hidden vectors is formed:

$$H^{(1)} = \{h_1, h_2, \dots, h_N\}. \quad (6)$$

Each vectorh_i contains not only the eigenfeatures of the corresponding node, but also aggregated information about its local graph neighborhood. This is why the subsequent generation of synthetic examples in the space$H^{(1)}$ is more accurate than interpolation in the original tabular feature space.

In the second stage, latent interpolation of minority nodes is performed. In the set$H^{(1)}$, a subset of nodes belonging to the illicit classV_{min} is identified. For each target node$h_i \in V_{min}$, one of the nearest neighborsh_j of the same class is determined, for example, by cosine similarity or Euclidean distance. A new synthetic node is formed according to the rule:

$$h_{syn} = h_i + \delta \cdot (h_j - h_i), \quad \delta \sim U(0,1), \quad (7)$$

where δ is a random coefficient uniformly distributed over the interval$[0,1]$. The resulting vectorh_{syn} is a synthetic latent representation of an illicit node located between two real examples of the minority class. The number of such nodes is determined by the target training set balancing coefficient.

In the third stage, the topological connections for the synthetic nodes are reconstructed. This stage fundamentally distinguishes Graph-SMOTE from classical SMOTE, since a synthetic node must be not only a new feature vector but also a full-fledged element of the graph structure. For this purpose, an edge predictor is used, which estimates the probability of a connection existing between the synthetic nodeh_{syn} and the real nodeh_v :

$$P(e_{syn,v}) = \sigma(h_{syn}^T W_{edge} h_v), \quad (8)$$

where W_{edge} is the training weight matrix of the edge predictor,$\sigma(\cdot)$ is the sigmoid activation function, and$P(e_{syn,v})$ is the probability of an edge existing between the synthetic and real nodes. Edges are added to the graph for those pairs of nodes for which the value of `$P(e_{syn,v})$` exceeds a specified threshold [2]. In this way, synthetic examples do not remain isolated but are included in the graph aggregation process.

The use of Graph-SMOTE allows for increasing the representation of the minority class without destroying the topological integrity of the transaction graph. Unlike simple duplication of illicit nodes or tabular feature interpolation, the proposed approach generates synthetic examples while taking into account the local structural context. This enhances the GATv2 model's ability to detect complex patterns of transaction obfuscation, including money laundering chains, asset mixing, and star-shaped transaction merging.

At the same time, using oversampling alone does not completely eliminate the imbalance problem, as the model may still favor the majority class during optimization. Therefore, the Weighted Cross-Entropy Loss (WCE) is additionally applied, which imposes different penalties for classification errors on the legitimate and illicit classes [5]:

$$L_{WCE} = -\frac{1}{N}\sum_{i=1}^{N}[w_1 y_i \log(\hat{y}_i) + w_0(1 - y_i)\log(1 - \hat{y}_i)], \quad (9)$$

where N is the number of nodes in the training mini-batch,$y_i \in \{0,1\}$ is the true label of a node,$\hat{y}_i \in [0,1]$ is the predicted probability of belonging to the illicit class,w_0 andw_1 are the weight coefficients for the legitimate and illicit classes, respectively.

The weight coefficients are determined inversely proportional to the number of examples in each class:

$$w_c = \frac{N_{total}}{C \cdot N_c}, \tag{10}$$

where N_{total} is the total number of training examples,C is the number of classes, andN_c is the number of examples in class$c$. This approach increases the penalty for misclassifying rare illicit nodes and reduces the probability of overlooking them.

To further enhance learning on complex examples, Focal Loss can be applied, which reduces the contribution of objects that are easy to classify and focuses optimization on nodes located near the decision boundary:

$$L_{Focal} = -\frac{1}{N}\sum_{i=1}^{N} \alpha_t \, (1 - \hat{p}_t)^{\gamma} \log(\hat{p}_t), \tag{11}$$

where $\hat{p}_t$ is the predicted probability of the true class,α_t is the class balance coefficient, andγ is the focusing parameter. When$\gamma = 0$, this function reduces to standard weighted cross-entropy, whereas increasingγ increases the model's focus on difficult and misclassified examples [11].

Thus, the combination of Graph-SMOTE, a weighted loss function, and the GATv2 dynamic attention mechanism forms a comprehensive approach to training the model under conditions of extreme class imbalance. Graph-SMOTE ensures a structurally sound expansion of the minority class, WCE increases the penalty for missing illicit transactions, and Focal Loss allows training to focus on the most complex cases of transaction obfuscation. It is precisely this combination that enhances the model's ability to detect rare but critically important threats in blockchain systems.

Comparison Methodology Approach

A comparative assessing of the proposed approach's effectiveness was conducted taking into account the specific characteristics of blockchain data: the graph structure of transactions, a high proportion of unlabelled nodes, and a significant imbalance between legitimate and illicit examples. The main criterion for the experiment's validity was ensuring identical conditions for all models: uniform preprocessing of features, the same data partitioning scheme, identical quality metrics, and averaging results across multiple runs.

Two representative on-chain datasets were used for experimental validation. The primary benchmark was the Elliptic Data Set, in which nodes are Bitcoin transactions and edges are asset flows between them. The dataset contains 203,769 transactions, grouped into 49 time slices, and 166 features for each node, including 94 local and 72 structural characteristics. This dataset was used for the main comparison of models in the task of classifying transactions into Licit and Illicit classes [1].

The comparison methodology involved a sequential comparison of four architectures: MLP as the baseline non-graph model, GCN as an isotropic graph convolutional network, GATv2 as a model with a dynamic attention mechanism, and GATv2 + Graph-SMOTE as a proposed configuration with topological oversampling of the minority class. This scheme allows for the separate assessing of the contributions of three components: the use of a graph structure, the application of anisotropic attention, and latent space balancing.

The comparison was performed on an independent test set, taking into account the temporal nature of the data, i.e., without mixing future transactions with training examples. To improve statistical reliability, the results were averaged over five runs with different initial values of the random number generator. This approach reduces the risk of accidental overestimation of metrics due to the peculiarities of a single data partition or model weight initialization.

Since the task of detecting transactional threats belongs to the class of tasks with extreme imbalance, the Accuracy metric was considered only as auxiliary. The primary quality metrics were Precision, Recall, and F1-score for the minority class Illicit. Precision characterizes the proportion of correct detections among all transactions marked by the model as risky; Recall reflects the mod-

el's ability to avoid missing actual threats; F1-score provides a comprehensive assessment of the balance between precision and recall.

The assessing metrics were determined using the following formulas:

$$Precision = \frac{TP}{TP+FP}, \tag{12}$$

$$Recall = \frac{TP}{TP+FN}, \tag{13}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \tag{14}$$

where TP is the number of correctly detected threats, FP is the number of false positives, and FN is the number of missed threats. For AML/CFT scenarios, the most critical factors are reducing FN and the F, since a false-negative decision means missing a potentially dangerous transaction or address.

Comparative Analysis Results

The analysis results show that switching from an MLP to a GCN yields a significant improvement in quality, because the model begins to account for the structure of connections. The subsequent transition to GATv2 improves detection completeness thanks to an anisotropic attention mechanism that dynamically identifies the most informative neighbours. GATv2 with Graph-SMOTE demonstrates the best balance between accuracy and recall, as latent oversampling reduces the shortage of minority class examples without disrupting the graph structure of the data.

Table 2

Comparative analysis of model performance

Model architecture	Precision	Recall	F1-score	Accuracy
MLP (Baseline)	0.6214	0.5830	0.6016	0.9420
GCN (Isotropic)	0.7582	0.7815	0.7697	0.9655
GATv2 (Multi-head)	0.8412	0.9105	0.8744	0.9780
GATv2 + Graph-SMOTE	0.8875	0.9022	0.8948	0.9845

The results confirm that the graph structure is highly informative for transaction threat detection tasks. The increase in the F1-score from 0.6016 for MLP to 0.7697 for GCN indicates that analysing isolated vectors is insufficient for blockchain data. GCN accounts for local connections, but its isotropic aggregation does not allow for fully distinguishing important edges from background ones.

The GATv2 model achieves an F1-score of 0.8744 and a Recall of 0.9105. This means that the attention mechanism significantly improves the ability to detect rare anomalous transactions. For AML/CFT tasks, a high Recall is particularly important, as a false-negative decision means missing a potential financial cyber threat. At the same time, simply increasing Recall without controlling Precision can lead to a large number of false alarms.

Integrating Graph-SMOTE allows us to increase Precision to 0.8875 and achieve the highest F1-score of 0.8948. This indicates that latent balancing improves not only anomaly detection but also the quality of class separation. In practice, this means a reduced workload for SOC analysts, as the model generates fewer false positives while maintaining the ability to detect dangerous patterns.

Another important point is that the proposed approach is compatible with subsequent decision explanation. GATv2 attention weights can be used to identify the most significant neighbouring nodes, while explainable AI methods can be used to assess the contribution of individual features to risk scoring [3]. This makes the model suitable not only for automatically flagging risky transactions but also for generating reasoned reports in digital forensics.

Discussion of Practical Application

The practical implementation of the method can be realized as a separate risk-scoring module within a cryptocurrency transaction monitoring system. As input, such a module receives an updated segment of the transaction graph, extracts local and structural features, constructs graph embeddings using GATv2, and determines the probability that a node belongs to a risk class. If the established threshold is exceeded, the transaction or address can be forwarded to an analyst for manual review.

This approach is most valuable in scenarios where risky transactions are rare, but their failure to be detected creates significant financial, regulatory, or reputational risks. Such scenarios include AML/CFT checks of cryptocurrency exchanges, monitoring of ransomware payments, detection of accumulator addresses, and analysis of suspicious asset withdrawal routes. In these conditions, latent oversampling allows for increased representation of rare patterns without mechanically duplicating training examples.

However, the method does not eliminate all limitations inherent in blockchain analytics. Classification quality depends on the reliability of the initial labelling, the completeness of the feature space, and the timeliness of the training data. Furthermore, attackers can adapt their behaviour to known monitoring mechanisms, alter the frequency of transfers, use new mixing services, or create fictitious intermediate addresses. Therefore, the model should be applied within a framework of periodic retraining and active expert involvement.

The best potential direction is to combine the proposed method with active learning. In this mode, transactions with high prediction uncertainty or atypical attention coefficients are forwarded to an analyst for manual verification. After confirmation, the new pattern can be added to the training set, allowing the system to respond to concept drift and the emergence of new obfuscation schemes.

Conclusions

This work presents an approach to assessing transaction threats in blockchain systems based on latent oversampling of graph embeddings. It is shown that the main problems of on-chain data analysis are extreme class imbalance, a significant proportion of unlabelled nodes, the topological complexity of the transaction graph, and the inadequacy of analysing isolated features.

The scientific result lies in the further development of the threat assessment method by integrating synthetic oversampling of the minority class directly into the latent space of GATv2 graph embeddings. Unlike common SMOTE, the proposed approach accounts for the structural connectivity of the graph and generates synthetic anomaly nodes in a space where the context of the transactional environment is already encoded.

A comparative analysis shows that the GATv2 + Graph-SMOTE configuration provides the best balance between Precision, Recall, and F1-score. Achieving an F1-score of 0.8948 confirms the effectiveness of combining dynamic graph attention with latent data balancing for the detection of rare transactional threats.

The practical value of the method lies in its applicability to AML/CFT systems, digital forensics, and SOC monitoring. Further research should focus on improving the interpretability of decisions, stress testing against adversarial attacks, optimizing inference on large graphs, and implementing active learning to counter concept drift.

References

1. Elliptic Data Set. URL: https://www.kaggle.com/datasets/ellipticco/elliptic-data-set.

2. Ma X., Wu J., Xue S., Yang J., Zhou C., Sheng Q.Z., Xiong H., Akoglu L. A Comprehensive Survey on Graph Anomaly Detection With Deep Learning // IEEE Transactions on Knowledge and Data Engineering. 2023. Vol. 35, No. 12. P. 12012–12038. DOI: https:// 10.1109/TKDE.2021.3118815.

3. Preuveneers D., Rimmer V., Tsingenopoulos I., Spooren J., Joosen W., Ilie-Zudor E. Chained Anomaly Detection Models for Federated Learning: An Intrusion Detection Case Study // Applied Sciences. publisher, 2018. Vol. 8, No. 12. DOI: https://10.3390/app8122663.

4. Shaizat M., Mussiraliyeva S. Enhanced identification of illicit bitcoin transactions through genetic algorithm-based feature selection // Eastern-European Journal of Enterprise Technologies. 2025. Vol. 4, 9 (136). P. 34–42. DOI: https://10.15587/1729-4061.2025.335630.
5. Aladhadh S., Alwabli H., Moulahi T., Asqah M.A. BChainGuard: A New Framework for Cyberthreats Detection in Blockchain Using Machine Learning // Applied Sciences. publisher, 2022. Vol. 12, No. 23. DOI: https:// 10.3390/app122312026.
6. Luša R., Pintar D., Vranić M., Luša R., Pintar D., Vranić M. TE-G-SAGE: Explainable Edge-Aware Graph Neural Networks for Network Intrusion Detection // Modelling. publisher, 2025. Vol. 6, No. 4. DOI: https:// 10.3390/modelling6040165.
7. Ashfaq T., Khalid R., Yahaya A.S., Aslam S., Azar A.T., Alsafari S., Hameed I.A. A Machine Learning and Blockchain-Based Efficient Fraud Detection Mechanism // Sensors. publisher, 2022. Vol. 22, No. 19. DOI: https://10.3390/s22197162.
8. Veličković P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y. Graph Attention Networks // International Conference on Learning Representations. 2018. DOI: https://10.48550/arXiv.1710.10903.
9. Brody S., Alon U., Yahav E. How Attentive are Graph Attention Networks? // International Conference on Learning Representations. 2021. DOI: https://10.48550/arXiv.2105.14491.
10. Zhang L., Chen W., Wang W., Jin Z., Zhao C., Cai Z., Chen H. CBGRU: A Detection Method of Smart Contract Vulnerability Based on a Hybrid Model // Sensors. Publisher. 2022. Vol. 22, No. 9. DOI: https:// 10.3390/s22093577.
11. Hwang S.-J., Choi S.-H., Shin J., Choi Y.-H. CodeNet: Code-Targeted Convolutional Neural Network Architecture for Smart Contract Vulnerability Detection // IEEE Access. 2022. Vol. 10. P. 32595–32607. DOI: https://10.1109/ACCESS.2022.3162065.

Надійшла до редколегії 02.07.2026
Прийняття до друку після рецензування 18.09.2026
Публікація (оприлюднення) 30.09.2026

Відомості про авторів:
Просолов Владислав Валерійович – Харківський національний університет радіоелектроніки, аспірант кафедри безпеки інформаційних технологій; Україна; e–mail: vladyslav.prosolov@nure.ua; ORCID: https://orcid.org/0009-0002-1276-4828
Халімов Геннадій Зайдулович – д-р техн. наук, професор, Харківський національний університет радіоелектроніки, завідувач кафедри безпеки інформаційних технологій; Україна; e-mail: hennadii.khalimov@nure.ua; ORCID: https://orcid.org/0000-0002-2054-9186