Article

Graph Neural Network and Increment Learning in Blockchain

Chang Qi *

Posted Date: 26 September 2024

doi: 10.20944/preprints202409.2117.v1

Keywords: Graph Neural Networks; Incremental Learning; Blockchain

Article

Graph Neural Network and Increment Learning in Blockchain

Chang Qi

Independent Researcher; changchannieqi1@gmail.com

Abstract: This paper investigates the integration of Graph Neural Networks (GNNs) and incremental learning for blockchain applications, with a focus on fraud detection, anomaly detection, and smart contract verification. By leveraging graph structure exploitation, GNNs can propagate information across nodes, reducing label dependency. Incremental learning enhances the model's adaptability to evolving blockchain networks, allowing continual learning without full retraining. Together, these technologies provide a scalable, efficient solution for improving security, performance, and adaptability in decentralized financial systems and smart contract environments.

Keywords: graph neural networks; incremental learning; blockchain

Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for analyzing complex and relational data structures, particularly in scenarios where data can be represented as graphs. This makes them highly applicable to blockchain technology, where decentralized systems operate with intricate networks of transactions, users, and smart contracts. This paper explores the synergies between GNNs and blockchain technology, providing a comprehensive understanding of how these two fields intersect. We will discuss the key characteristics of GNNs, their ability to handle graph-structured data, and how they can be applied to blockchain networks to address various challenges, such as scalability, security, and fraud detection.

Literature Review

Graph Neural Networks (GNNs) extend traditional deep learning models by directly operating on graph data, which consists of nodes (representing entities) and edges (representing relationships between entities). Unlike conventional neural networks that operate on grid-like data, such as images or sequences, GNNs are well-suited for non-Euclidean data, which is often the case in real-world scenarios like social networks, molecular structures, and, notably, blockchain networks. GNNs leverage the relational structure of data to learn node representations, making them powerful for tasks like node classification, link prediction, and graph classification. GNN can also assist in sports analytics demonstrated in [1].

A key strength of GNNs lies in their ability to capture both the local and global structure of graphs [2]. Through message-passing mechanisms, nodes in a graph update their representations by aggregating information from their neighbors. This iterative process enables GNNs to learn rich and contextualized node embeddings that encapsulate the relationships within the graph. These embeddings are crucial for downstream tasks like anomaly detection, fraud detection, and network analysis, which are particularly relevant in the blockchain domain.

Blockchain is a distributed ledger technology that enables secure and transparent record-keeping without relying on a central authority. Each transaction or data point in a blockchain is stored in a block, which is linked to other blocks, forming a chain. This chain of blocks is maintained across a decentralized network of nodes, ensuring the integrity of the data through consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS).

 doi:10.20944/preprints202409.2117.v

While blockchain has gained immense popularity due to its applications in cryptocurrencies like Bitcoin and Ethereum, its potential extends far beyond financial systems. Blockchain can be used in supply chain management, healthcare, voting systems, and more. However, the scalability and complexity of blockchain networks pose significant challenges, particularly as these networks grow in size and complexity. This is where GNNs come into play, offering a novel approach to analyzing and optimizing blockchain networks.

Several key studies have explored incremental learning in GNNs and blockchain networks. [3] discuss incremental updates to neural networks in blockchain frameworks, focusing on optimizing resource use in decentralized environments. [4] pioneers in GNN architecture, provide a foundation for applying incremental learning to GNNs in evolving graph-based structures like blockchain.

Applications of GNNs in Blockchain

GNNs offer a unique perspective for addressing various challenges in blockchain technology. Below, we outline some of the key applications of GNNs in this domain.

Blockchains are susceptible to fraudulent activities, including double-spending attacks, Sybil attacks, and malicious transactions. Given the relational nature of blockchain transactions, GNNs are well-suited for detecting anomalous behavior within the network. By representing blockchain transactions as graph data, where nodes represent users and edges represent transactions, GNNs can be trained to identify suspicious patterns or outliers that may indicate fraud. We can deploy GNN client that learns from previous patterns and give a level of confidence to the newly discovered patterns in blockchain. Alternatively, we can use an influence function to measure the fact and the outcome via GNN as illustrated by [5].

Traditional fraud detection methods may rely on heuristic rules or statistical models, which may not scale well to complex blockchain networks. GNNs, on the other hand, can automatically learn patterns from the data and identify fraudulent activities with higher accuracy. Moreover, as blockchain networks evolve, GNNs can adapt to changes in the network structure and detect emerging forms of fraud.

Blockchain networks can be highly complex, with numerous participants interacting through transactions. Analyzing these networks to identify communities of users or clusters of similar behavior can provide valuable insights into the dynamics of the blockchain. GNNs are particularly effective in community detection tasks, where the goal is to identify groups of nodes (users) that exhibit similar transactional behavior.

Community detection in blockchain networks can be useful for various purposes, including understanding user behavior, optimizing consensus mechanisms, and enhancing network security. For example, identifying clusters of malicious nodes can help in mitigating the impact of attacks on the network. GNNs can also be used to predict new relationships or interactions within the network, facilitating the development of more efficient and secure blockchain systems.

One of the most pressing challenges in blockchain technology is scalability. As the number of users and transactions in a blockchain network grows, the computational and storage requirements increase exponentially. GNNs can help optimize the structure and efficiency of blockchain networks by identifying bottlenecks, optimizing consensus mechanisms, and reducing redundant transactions.

By analyzing the topology of blockchain networks, GNNs can identify critical nodes or edges that play a central role in maintaining the network's integrity. Optimizing the connectivity of these nodes can improve the overall performance of the network, making it more scalable and resilient to attacks. Furthermore, GNNs can be used to develop more efficient consensus algorithms that reduce the time and computational resources required to validate transactions.

Smart contracts are self-executing contracts with the terms of the agreement directly written into code. While smart contracts offer significant advantages in terms of automation and trust, they also introduce potential vulnerabilities. Bugs or exploits in smart contracts can lead to severe financial losses, as demonstrated by several high-profile incidents in the blockchain space. Smart Contract Analysis and Verification refers to the process of ensuring that smart contracts, which are self-executing agreements with predefined rules encoded in blockchain, function as intended without

 doi:10.20944/preprints202409.2117.v

bugs, vulnerabilities, or logic errors. Using Graph Neural Networks (GNNs) in combination with blockchain analysis, smart contract verification can be enhanced by detecting patterns and anomalies in code execution paths or user interactions.

GNNs can be applied to analyze the structure and execution of smart contracts, identifying potential vulnerabilities or inefficiencies. By representing the smart contract code and its interactions as a graph, GNNs can learn to detect patterns that may indicate security flaws or performance bottlenecks. This can enhance the security and reliability of smart contracts, reducing the risk of exploitation.

GNNs can model the flow of data and relationships between smart contract components as a graph, allowing for more efficient detection of errors, particularly when paired with incremental learning methods. As smart contracts evolve or are upgraded, incremental learning enables the model to continually learn new patterns and interactions without having to relearn from scratch. This ensures the verification process adapts to dynamic contract environments.

In practice, the verification process identifies logic flaws, security vulnerabilities, or backdoors that might compromise the smart contract. Blockchain platforms such as Ethereum rely heavily on smart contracts, and securing these contracts is paramount. GNNs, with their capacity to exploit graph structures, can effectively analyze the interdependencies between contracts, transactions, and users, enhancing the reliability and security of blockchain ecosystems. Several studies have emphasized the importance of smart contract verification, such as work by [6], which focused on security vulnerabilities in Ethereum smart contracts. GNNs, when combined with active learning strategies, can prioritize the most uncertain or risky components of smart contracts for deeper analysis, enhancing the efficiency and accuracy of the verification process in evolving blockchain environments.

Uncertainty Sampling with GNN to Blockchain

Uncertainty sampling is an active learning technique where the model focuses on the data points about which it is least confident in its predictions. In Graph Neural Networks (GNNs), uncertainty sampling helps enhance learning efficiency by selecting nodes with the highest prediction uncertainty for manual labeling. This approach prioritizes learning from the most informative nodes, thus improving model accuracy with fewer labeled instances. When applied to blockchain networks, uncertainty sampling efficiently handles the dynamic and complex nature of transactions, aiding in tasks such as fraud detection and anomaly detection.

In practice, the GNN estimates uncertainty by evaluating prediction probabilities across possible labels, identifying nodes with ambiguous predictions (e.g., those with low confidence scores or high entropy). These uncertain nodes are then selected for labeling by an oracle, which provides true labels for the model to learn from. The model is updated iteratively with the newly labeled data, improving its predictive power, especially in uncertain or evolving network scenarios.

This method is particularly valuable in environments where acquiring labeled data is costly or time-consuming, like blockchain networks, where constant changes occur due to new transactions or evolving fraud patterns. With uncertainty sampling, models can focus on the most ambiguous areas of the network, refining predictions in regions where new or anomalous behavior may emerge. This method thus maximizes efficiency in blockchain-related applications such as security, anomaly detection, and network analysis.

The concept of uncertainty sampling traces back to the foundational work of [7,8], and has been further developed for use in machine learning and graph-based models by [9]. These techniques are especially useful in settings with complex graph structures, such as blockchain systems. Contrastive learning can be added at the end of GNN implementation to enforce the current learning with a negative output study sample, this proves to be useful in [10].

Graph structure exploitation in GNN

Graph structure exploitation in Graph Neural Networks (GNNs) is the process of utilizing the inherent connections between nodes to propagate information, thereby minimizing the need for

 doi:10.20944/preprints202409.2117.v

extensive labeled data. In a graph, such as a blockchain network, the relationships between nodes (e.g., users, transactions) are crucial for learning. GNNs achieve this by aggregating information from neighboring nodes through techniques like message passing and convolutions, which allow the model to infer patterns from unlabeled data. This enables nodes with limited labels to gain information from their connected neighbors, reducing reliance on labeled data while improving learning efficiency.

In blockchain applications, this ability to propagate information is critical. For example, if a few nodes (transactions or users) are labeled as fraudulent, the model can spread that information across the graph to detect similar, unlabeled instances. Techniques like Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) enhance this process by prioritizing relevant connections and ensuring that both local and global graph structures are considered. By doing so, GNNs can generalize patterns from sparse labeled data, making them effective for dynamic and complex systems like blockchain networks.

Several studies have explored how GNNs can leverage graph structures effectively. Notably, [4] pioneered semi-supervised GCNs, while [11] introduced attention mechanisms in GNNs to focus on critical node relationships. These approaches have revolutionized graph-based learning, especially in environments like blockchain, where labeling every data point is costly and inefficient. By exploiting graph structures, GNNs help ensure accurate predictions and insightful analyses, even with limited labeled data.

Incremental Learning and GNN for Blockchain

Integrating incremental learning into Graph Neural Networks (GNNs) for blockchain applications can significantly enhance adaptability in dynamic environments. Incremental learning, or continual learning, enables models to update knowledge as new data arrives without retraining from scratch. This is crucial for blockchain networks, which continually grow as new nodes and transactions are added.

In dynamic blockchain networks, incremental learning allows GNNs to update node representations and edge relationships efficiently. As blockchain networks expand with new blocks and transactions, GNNs using incremental learning can quickly adjust to new patterns while retaining previously learned knowledge. This continuous adaptation is vital in blockchain applications, where the network evolves rapidly, and scalability is a significant concern. Several methods have been proposed in literature to implement incremental learning in GNNs. For example, [12] proposed gradient episodic memory (GEM), a method that helps GNNs preserve prior knowledge while learning new tasks. This technique can be applied to blockchain networks, ensuring that GNNs retain their understanding of historical transactions while updating to incorporate new data.

Blockchain networks are prone to evolving fraudulent schemes, and incremental learning allows GNN models to detect new patterns of fraud while retaining the knowledge of previously known schemes. Smart contracts may evolve over time, with incremental learning enabling GNNs to adapt to changes in the contract structure, optimizing security and efficiency. As blockchain networks grow, incremental learning can help maintain efficient performance by updating the GNN only as needed, avoiding the computational cost of complete retraining.

Incorporating influence functions into incremental learning enhances the accuracy of Graph Neural Networks (GNNs) for blockchain applications by evaluating the impact of new data on the model's predictions. Influence functions allow the model to trace how new nodes, edges, or transactions affect learning, enabling the selective integration of valuable data while preserving performance. This is especially beneficial in blockchain networks, where constant updates occur. By prioritizing high-impact data, GNNs maintain accuracy in tasks like fraud detection and smart contract verification, ensuring efficiency as the blockchain evolves.

In GNNs, influence functions specifically monitor how new transactions or nodes impact connected entities, helping the model adapt effectively without retraining from scratch. This process improves model reliability, particularly in detecting changes or anomalies within blockchain

 doi:10.20944/preprints202409.2117.v

networks, such as new fraudulent activity or alterations in smart contracts. By minimizing the influence of less significant data and focusing on critical updates, models can continually improve without losing accuracy, making influence functions an essential tool for maintaining GNN performance in evolving blockchain systems. We can use generative AI (XAI) as shown in [13] to connect the incremental learning with blockchain such that the influence function can be recognized in both the GNN and blockchain system.

Conclusion

In conclusion, integrating Graph Neural Networks (GNNs) with blockchain technology offers transformative potential in various applications, from fraud detection to smart contract analysis and verification. By leveraging graph structure exploitation, GNNs can efficiently propagate information across blockchain networks, minimizing the need for extensive labeled data. The incorporation of incremental learning further enhances adaptability, allowing GNN models to stay relevant as blockchain networks grow and evolve. This combination of GNNs and blockchain creates a robust framework for improving security, scalability, and transaction efficiency, opening new frontiers for financial applications and decentralized systems.

References

1. Z. Wang, Y. Zhu, Z. Li, Z. Wang, H. Qin and X. Liu, "Graph neural network recommendation system for football formation," *Applied Science and Biotechnology Journal for Advanced Research,* vol. 3, p. 33–39, 2024.
2. W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang and J. Tang, "Graph structure learning for robust graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020.
3. C. Zeno, I. Golan, E. Hoffer and D. Soudry, "Task-agnostic continual learning using online variational bayes with fixed-point updates," *Neural Computation,* vol. 33, p. 3139–3177, 2021.
4. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907,* 2016.
5. J. Wu, Y. Yang, Y. Qian, Y. Sui, X. Wang and X. He, "Gif: A general graph unlearning strategy via influence function," in *Proceedings of the ACM Web Conference 2023*, 2023.
6. L. Brent, N. Grech, S. Lagouvardos, B. Scholz and Y. Smaragdakis, "Ethainter: a smart contract security analyzer for composite vulnerabilities," in *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2020.
7. L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," *arXiv preprint arXiv:1803.08533,* 2018.
8. F. Wang, Y. Liu, K. Liu, Y. Wang, S. Medya and P. S. Yu, "Uncertainty in Graph Neural Networks: A Survey," *arXiv preprint arXiv:2403.07185,* 2024.
9. D. B. Markant, B. Settles and T. M. Gureckis, "Self-directed learning favors local, rather than global, uncertainty," *Cognitive science,* vol. 40, p. 100–120, 2016.
10. Z. Li, B. Wang and Y. Chen, "A Contrastive Deep Learning Approach to Cryptocurrency Portfolio with US Treasuries," *Journal of Computer Technology and Applied Mathematics,* vol. 1, pp. 1-10, 2024.
11. P. Veličković, "Message passing all the way up," *arXiv preprint arXiv:2202.11097,* 2022.
12. R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in neural information processing systems,* vol. 32, 2019.
13. Q. Xu, Z. Feng, C. Gong, X. Wu, H. Zhao, Z. Ye, Z. Li and C. Wei, "Applications of explainable AI in natural language processing," *Global Academic Frontiers,* vol. 2, p. 51–64, 2024.