On the Use of Heterogeneous Graph Neural Networks for Detecting Malicious Activities: a Case Study with Cryptocurrencies

Stefano Ferretti
Department of Pure and Applied Sciences, University of Urbino
Urbino, Italy
stefano.ferretti@uniurb.it

Gabriele D'Angelo
Department of Computer Science and Engineering, University of Bologna
Bologna, Italy
g.dangelo@unibo.it

Vittorio Ghini
Department of Computer Science and Engineering, University of Bologna
Bologna, Italy
vittorio.ghini@unibo.it

ABSTRACT

This paper presents a study on the application of Heterogeneous Graph Neural Networks (HGNNs) for enhancing the security of complex social systems by identifying illicit and malicious behaviors. We focus on digital asset tokenization, a key component in the construction of many innovative social services, with the aim of classifying token exchanges and identifying illicit activities. Utilizing the Elliptic++ dataset, we demonstrate the efficacy of HGNNs in identifying illicit activities in token-based exchanging applications. In particular, we evaluate four different HGNN architectures, i.e. Heterogeneous GAT, Heterogeneous SAGE, HGT (Heterogeneous Graph Transformer), and HAN (Heterogeneous Attention Network). Our results underscore the importance of characterizing and describing interactions in these complex systems, both for studying the system dynamics and for activating mechanisms to cope with cybersecurity issues, like misuses and usurpation of resources in social systems.

CCS CONCEPTS

• **Security and privacy → Human and societal aspects of security and privacy**; • **Applied computing → Investigation techniques**; • **Computing methodologies → Machine learning approaches**; **Neural networks**.

KEYWORDS

Graph Neural Networks, Heterogeneous Graphs, Blockchain, Anti Money Laundering

ACM Reference Format:
Stefano Ferretti, Gabriele D'Angelo, and Vittorio Ghini. 2024. On the Use of Heterogeneous Graph Neural Networks for Detecting Malicious Activities: a Case Study with Cryptocurrencies. In *4th International Workshop on OPEN CHALLENGES IN ONLINE SOCIAL NETWORKS (OASIS '24), September 10–13, 2024, Poznan, Poland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3677117.3685009

OASIS '24, September 10–13, 2024, Poznan, Poland

ACM ISBN 979-8-4007-1082-7/24/09
https://doi.org/10.1145/3677117.3685009

1 INTRODUCTION

Social systems are a typical example of a system that can be represented as a complex network. The complexity primarily comes from the large number of entities (i.e., individuals) that can be involved. This generates patterns and structures whose analysis can only be carried out by treating these networks with appropriate mathematical modeling tools. Moreover, the introduction of novel communication technologies, types of devices, services and applications make interactions in the social system much more complex and heterogeneous than before.

As an example, take the problem of designing novel digital services to promote the creation of smart territories [7]. In this case, the idea is to make good use of social interactions by providing means for the sharing of resources, services and data, through the use of rewarding systems [12]. This allows to create a substrate of micro-services that, when combined, allow for the development of more sophisticated smart services. All of this often needs to occur in a completely decentralized context, where peer-to-peer technologies enable self-organization and service combination. Blockchain technologies, smart contracts, and token systems based on cryptocurrencies are usually at the basis of the development of decentralized applications. On one hand, this decentralized approach is based on the complete absence of a single controlling entity, thus freeing the services from having a trusted third party that coordinates (and controls) everything. This solves several issues strictly related to the presence of a single point of failure: single point of trust, privacy, surveillance, and self sovereignty. On the other hand, it raises some relevant security concerns. In fact, the lack of complete knowledge and control of the system opens the way to malicious behaviors that try to exploit vulnerabilities in the system, usurpation of resources and services, and then misuses.

With the aim to to establish a healthy and collaborative system where shared assets represent exchange goods that promote social good, it becomes important to identify the malicious actors and illicit activities within such a complex and heterogeneous digital ecosystem. This not only enhances the efficiency and reliability of these systems but also promotes accountability and trust among its users.

In the attempt to characterize the activities and interactions among different entities and system components, the use of heterogeneous graphs, as a modeling tool, can be highly beneficial. A heterogeneous graph, also known as a multigraph, is a graph in which nodes and edges can be of different types. This allows for a useful and very descriptive representation of complex systems,

where different types of entities (nodes) can interact in various ways (edges). For instance, in a social network, nodes could represent users and organizations, while edges could represent different types of relationships such as friendship, following, or membership. In the context of digital ecosystems, these graph-based approaches can provide valuable insights into the dynamics of the system. They allow to model and analyze interconnected relationships between various entities and activities. This can aid in identifying patterns, classify interactions, as well as predicting future interactions. Furthermore, new machine learning techniques for the analysis of heterogeneous graphs have been recently proposed. In particular, Heterogeneous Graph Neural Networks (HGNNs) are a type of neural network designed to handle graph structured data, where nodes and edges can be of different types [22]. They enable the classification of various entities and their interrelationships, utilizing combined knowledge derived from the complex heterogeneous graph.

In this paper, we present a study aimed at assessing whether the use of HGNNs can indeed enhance the security of a complex social system by identifying illicit and malicious behaviors. In particular, we focus on digital asset tokenization, as the underlying tool for the construction of social services, with the aim of classifying token exchanges and then identify illicit activities. Due to the lack of a specific (and publicly available) use-case dataset, in our analysis we use a Bitcoin transaction dataset. Despite Bitcoin not being an ideal candidate for resource sharing systems, its network structure is similar to other token-based digital systems. Hence, we believe that our classification study could serve as a preliminary validation for potential broader applications of our framework. More specifically, we utilized Elliptic++ as the dataset to build the heterogeneous graph [6]. It is an enhanced version of the Elliptic dataset, a large, labeled dataset of Bitcoin transactions, specifically designed for detecting illicit activities in Bitcoin. It includes features extracted from the blockchain data, as well as labels indicating whether each transaction is licit or illicit. The Elliptic++ dataset extends the Elliptic dataset by including over 822k Bitcoin wallet addresses, each with 56 features, and 1.27M temporal interactions.

The results show that with the help of heterogeneous graph analysis and the use of HGNNs, it is indeed possible to identify illicit activities in token based exchanging applications. This may contribute to the development of mechanisms that make the use of tokens on decentralized architectures a suitable tool for the development of smart services. Furthermore, the results demonstrate how important it is to be able to characterize and describe the interactions in these complex systems. This is done in order to study the dynamics of the system on one hand, and to activate mechanisms to cope with misuses and usurpation of resources in social systems.

The remainder of this paper is organized as follows. Section 2 provides the background needed for the rest of the paper. Section 3 describes the methodology for the study, thus describing the employed dataset, the classifiers and the considered metrics. Section 4 is focused on the results obtained using different GNN classifiers. Finally, Section 5 provides some concluding remarks.

2 BACKGROUND

This section provides the necessary background on the use of heterogeneous graphs to model complex social systems, as well as an introduction to the Heterogeneous Graph Neural Networks (HGNNs).

2.1 Heterogeneous Graphs

Relationships among entities can be seen as a network [1]. This is true in a variety of different domains, ranging from social networks, web of things, biological networks, up to economic applications. In many cases, the involved entities and types of interactions are of the same type. For instance, in a social network, entities are individuals while the interactions represent friendship between people. These kinds of interactions can be easily represented as a graph, whose study often permits to identify interesting properties and patterns of the system being modeled.

In reality, real-world entities and their interactions often exhibit multiple types, forming heterogeneous graphs with rich structural and semantic information [21]. These heterogeneous graphs arise in various domains, such as protein interaction networks, social networks, traffic networks, academic collaboration networks, and knowledge graphs [2, 4, 11, 13, 24]. Thus, in many cases there is the need to represent and connect different types of entities and relations. For instance, we might consider a set of individuals whose relationships might signify friendship, collaboration, mentorship, or just casual encounters. All this can be represented using an heterogeneous graph. The key aspects of such a graph are i) node diversity - each node is a specific type with distinct attributes; ii) edge diversity and variability - edges connect nodes, but they represent a different type of relation, usually characterized by a certain type and attributes as well.

Clearly enough, this richer view of complex graph allows capturing nuances and multiple aspects of a system, that are connected together in a single object (i.e., the graph). The study of these graphs thus requires the use of adequate methodologies for their analysis.

A key concept at the basis of an heterogeneous graph is the meta-relation, used to denote the type of relationship between two nodes of a certain type. For an edge e linked from source node s to target node t, its-meta relation is denoted as $\langle \tau(s), \phi(e), \tau(t) \rangle$. Here, $\tau(s)$ and $\tau(t)$ represent the types of the source and target nodes, respectively, and $\phi(e)$ represents the type of the edge. For instance, in an academic network a meta-relation can identify the fact that an author wrote a specific paper. In this case, the meta relation would be ⟨Author, Writes, Paper⟩ that, to simplify the notation, can be informally represented as "Author-Paper", being the type of relation implicitly identified, in this specific case. Meta-relations allow the model to maintain dedicated representations for different types of nodes and edges, thereby effectively capturing the heterogeneity of the graph.

Then, a meta-path is defined as a sequence of meta-relations. For instance, in an academic network, a meta-path could be "Author-Paper-Author" (APA), which represents the co-author relation between two authors. Another example could be "Author-Paper-Venue-Paper-Author" (APVPA), which represents the relation between two authors who have published papers in the same venue.

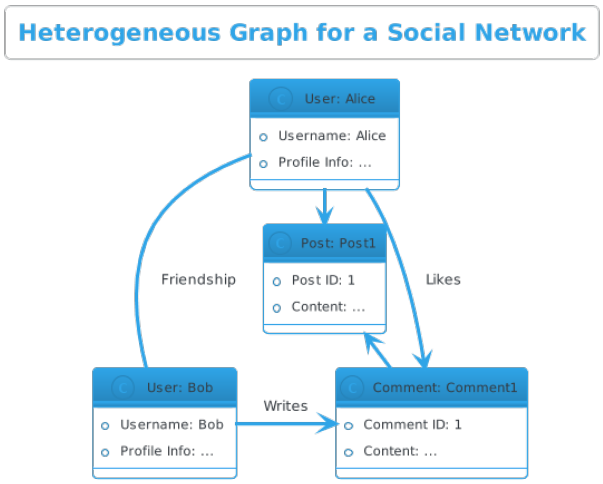

Figure 1: An example of a social network represented as a heterogeneous graph.

In the following, we provide some examples where heterogeneous graphs can be used to model social interactions in different system applications.

2.1.1 Social Media Networks. Heterogeneous graphs offer a powerful framework for modeling social media networks, as they can represent diverse entities and their interrelationships. In the context of social media, entities can be users, posts, comments, or even reactions, while interactions can represent friendships, follows, likes, shares, or comments [14]. This multiplicity of connections captures the rich and complex nature of social media networks. Furthermore, these graphs can be dynamic, reflecting the temporal nature of social media interactions [10].

Figure 1 shows a simple example demonstrating how a social network can be represented as a heterogeneous graph. In this graph, Alice and Bob are users with their features included. Alice creates a post (Post1), while Bob comments on that post (Comment1), and Alice likes the comment.

Recent advancements in machine learning, particularly HGNNs, have further enhanced our ability to analyze these heterogeneous graphs, enabling us to uncover hidden patterns and insights in social media data.

2.1.2 Smart Territories. Heterogeneous graphs are an essential tool for modeling complex systems such as smart cities and smart territories [2, 11, 24]. In fact, these graphs can capture the intricate relationships among various entities, such as people, devices, services, and infrastructures. Thus, heterogeneous graphs can be of help in providing a comprehensive and nuanced understanding of smart cities and territories, thereby facilitating more effective management and decision-making processes. For instance, [11] discusses the use of knowledge graphs, a type of heterogeneous graph, for managing informed consent in smart cities. Similarly, in [2] a Semantic Knowledge Based Graph (SKBG) model is proposed to handle the large-scale, multi-source, and complex data generated in smart cities. In terms of smart territories, GeoHG was introduced in [24], a heterogeneous graph structure for learning comprehensive region embeddings for various downstream tasks.

2.1.3 Cybersecurity. Heterogeneous graphs can be used in the field of cybersecurity [4, 13]. In fact, heterogeneous graphs can capture the relationships among various entities in a computer system such as processes, files, and sockets [4]. Thus, they can help in identifying threats and enhancing the performance of intrusion detection and prevention mechanisms. For instance, [13] highlighted how knowledge graphs can aggregate and represent complex data, thereby supporting security decision-making and predicting warnings. Other works presented graph-based models to infer malicious domains. In [5], a heterogeneous knowledge base was built using cyber threat intelligence. Then, authors formalize threat attribution as a link prediction task on the heterogeneous graph and propose a graph embedding method to extract features of attackers. In [23], a heterogeneous information network was used to model the usage of the DNS protocol while considering different types of nodes, and different types of relation (such as request and resolution messages). Then, a HGNN model was used to detect malicious domains in a semi-supervised learning approach.

2.1.4 Tokenizing circularity in agri-food systems. The Circular Economy (CE) has been considered as a sustainable economic system in which resource depletion is maximally prevented. CE is intrinsically linked to the concept of cleaner production as they both aim at preventing the production of waste, while increasing efficiencies in the uses of energy, water, resources, and human capital. The concept of circularity is an important theme in research on agri-food systems as they are heavy users of natural resources and suppliers of essential goods and services. For many parties in the agri-food supply chains, a pressing issue is, however, how to trust or verify such claims of circularity as it is not always visible where a product comes from and what it is made of. As reported in [18], tokenization appears relevant to the circular economy in three ways: 1) enhancing traceability of physical and digital objects in supply chains; 2) improving transparency and credibility of circularity claims; 3) facilitating collaborative business ecosystems with incentives for more circular production and distribution. As circular supply chains are complex networks with closed loops of inputs, resources and outputs, the tracking and tracing of these highly interdependent flows is conceptually complex and operationally challenging. Thus, heterogeneous graphs can be of help in modeling the token interaction of agri-food systems and providing a global and detailed understanding, facilitating the governance system.

2.1.5 Combating Illicit Misuses of Cryptocurrencies. Cryptocurrencies can be modeled as a transaction graph, in various ways. In fact, in this type of economic system, wallets interact by exchanging tokens. Moreover, in the so called Unspent Transaction Output (UTXO) blockchain models, such as Bitcoin, also transactions can be linked together, since in each novel transaction, an address must spend some tokens which have been earned in a previous transaction. This representation as a network allows us to make diverse analyzes. For instance, user behaviours can be monitored and classified in a sort of "know your customer" perspective, even if addresses are pseudonymous.

But not only, it is well known that the decentralization of the blockchain technology opens the way for illicit activities. Tokens, which are digital assets that reside on their own blockchains, can be misused for money laundering purposes. The complexity and global nature of the cryptocurrency transaction graph can make such illicit activities challenging to detect and trace. However, recent studies demonstrated that the analysis of the transaction graph through deep learning techniques can lead to the classification of transactions and to the identification of illicit ones [15, 16]. These works used a dataset that allows for the construction of a homogeneous graph. In this graph, each node represents a transaction, and each edge (u, v) represents the fact that in transaction v, Bitcoins previously earned in transaction u are spent.

In this work, as a use case, we consider an extension of this use case applied to a dataset that allows for the creation of a heterogeneous graph, with two types of nodes, i.e., transactions and wallets. The edges are also heterogeneous, as they can connect nodes of different types.

2.2 Heterogeneous Graph Neural Networks

Graph Neural Networks (GNNs) have garnered significant attention in recent years for their ability to learn expressive node representations from graph-structured data [15]. Most GNNs focus on homogeneous graphs, containing only a single type of nodes and edges.

Heterogeneous Graph Neural Networks (HGNNs) address this complexity by extending GNNs to handle heterogeneous graphs [22]. These deep learning models can capture high-order nonlinear interactions among different types of nodes and edges. Various architectures for HGNNs have been proposed already, including attention mechanisms and multi-layer structures [3].

2.2.1 Heterogeneous Attention Network (HAN). Heterogeneous Graph Attention Networks (HAN) introduce a hierarchical attention mechanism, which includes node-level and semantic-level attentions [20]. At the core of the approach lie the previously mentioned concepts of meta-relation and meta-path.

Here, a meta-path is used to sample the heterogeneous neighbors of each node in the network. The node-level attention in HAN aims to learn the importance between a node and its meta-path based neighbors. The semantic-level attention, on the other hand, is able to learn the importance of different meta-paths. With the learned importance from both node-level and semantic-level attention, the HAN generates node embeddings by aggregating features from meta-path based neighbors in a hierarchical manner.

2.2.2 Heterogeneous Graph Transformer (HGT). The Heterogeneous Graph Transformer (HGT) is one of the first examples of GNNs working on heterogeneous data. To model heterogeneity, HGT introduces node- and edge-type dependent parameters to characterize a heterogeneous attention over each edge [9].

Not all relationships contribute equally to the target nodes. Therefore, HGT uses attention over features of each node and edge type. HTG extends the idea employed in Graph Attention Networks (GAT) [19]. In GAT, the graph structure is used to learn the representation of each node by considering the representation of its neighbors, similarly to a message passing approach, in which each node contributes by passing its representation to nodes it is connected. The difference between GAT and other GNNs, is that an attention mechanism is used to weight the contribution of neighbors. In HTG, this weight is calculated by using knowledge of the specific meta-relation involving connected nodes. Thus, the general significance of each meta-relation triplet is used to capture both the common and specific patterns of different relationships. In other words, meta-relations of the same type share some parameters, so that knowledge related to a specific type of interaction among two specific types of nodes can be transferred to other interactions of the same type.

Moreover, HGT is equipped to handle dynamic heterogeneous graphs. It introduces the relative temporal encoding technique, which is capable of capturing the dynamic structural dependency with arbitrary duration.

2.2.3 Heterogeneous Convolution Wrapper. A practical tool to generate a HGNN is the HeteroConv operator, available in the Pytorch library [17]. HeteroConv is a generic convolutional wrapper that can be used to deploy operators, originally designed for homogeneous graphs, to handle heterogeneous ones. As already mentioned, in a heterogeneous graph, different types of nodes and edges exist, and the HeteroConv wrapper allows for the definition of custom heterogeneous message and update functions to build arbitrary message passing GNNs for these graphs. The HeteroConv wrapper operates by passing messages from source nodes to target nodes based on the bipartite GNN layer given for a specific edge type. If multiple relations point to the same destination, their results will be aggregated according to a specified aggregation method.

In this work, we use this operator to apply SAGE convolutional operators to the heterogeneous graph [8], as well as GAT convolutional layers [19].

3 METHODOLOGY

In this section, we present a study that utilizes HGNNs for classification in social systems. Specifically, we take the problem of anti-money laundering on cryptocurrencies as our use case. We then introduce the dataset used, the employed classifiers and the metrics that we evaluate to assess their performance.

3.1 The Dataset

We have already mentioned that one of the fundamental aspects in a social interaction system is the ability to use a secure system for token exchange, aiming to minimize potential misuses of these tokens for illicit activities. The study we are conducting is directed towards this main goal. However, at present, we do not have a dataset related to a token-based system in a specific use case. To overcome this limitation, we have decided to use a dataset that refers to Bitcoin transactions. We are aware that the Bitcoin cryptocurrency is not a good candidate to be used as a token exchange instrument in resource sharing systems. However, as a cryptocurrency, its structure is similar to that of any other token-based digital system, in which transfer of tokens can be represented as a transaction graph [16]. Therefore, we believe that a classification study conducted on this cryptocurrency could also be valid for a preliminary validation of the framework as a potential system to be used in broader contexts.

The dataset on which the experiments were conducted is the Elliptic++ dataset [6], an extended version of the Elliptic dataset that contains information about Bitcoin transactions and related wallet addresses [15].

Table 1 shows the information about the transactions, while Table 2 reports information about the wallets. The dataset comprises 203,769 transactions and 822,942 wallet addresses, with each transaction and wallet address characterized by 183 and 56 features, respectively. Both transactions and wallets have a class label (Illicit, Licit, or Unknown). The dataset captures the temporal dynamics of the Bitcoin network with data spanning 49 time steps. The dataset is also characterized by a significant number of edges that represent three types of interactions: money flow between transactions, temporal interactions between wallet addresses and transactions, and interactions between wallet addresses. In particular, there are 234,355 "money flow" edges between transactions, 2,868,964 "addr-addr" edges between wallet addresses, and 1,314,241 "addr-tx-addr" edges representing interactions between wallet addresses and transactions. In terms of class distribution, the dataset is quite unbalanced, since it contains 4,545 illicit transactions and 14,266 illicit wallet addresses, as well as 42,019 licit transactions and 251,088 licit wallet addresses. A large number of transactions and wallet addresses are labeled as unknown (157,205 transactions and 557,588 wallet addresses), reflecting the challenges in obtaining ground truth labels in such networks.

Table 1: Summary of the Transactions in the Elliptic++ Dataset

Description	Count
Nodes (transactions)	203,769
Edges (money flow)	234,355
Time steps	49
Illicit (class-1)	4,545
Licit (class-2)	42,019
Unknown (class-3)	157,205
Features	183

Table 2: Summary of the Wallet Addresses in the Elliptic++ Dataset

Description	Count
Wallet addresses	822,942
Nodes (temporal interactions)	1,268,260
Edges (addr-addr)	2,868,964
Edges (addr-tx-addr)	1,314,241
Time steps	49
Illicit (class-1)	14,266
Licit (class-2)	251,088
Unknown (class-3)	557,588
Features	56

During the performed classification analysis, the dataset has been split into three subsets: a training set, a validation set, and a test set. The training set comprised 65% of the dataset, while the validation and test sets each contained 15% and 20% of the data, respectively.

3.2 The Employed Classifiers

The classifiers employed in this experiment are HGNNs, classifiers which are designed to tackle graphs containing nodes and edges of different kinds, with varying numbers of features. In this analysis, we used AdamW as optimizer. Each model was trained for a total of 1000 epochs. A fixed learning rate of 0.001 was set.

We compare four different types of HGNNs.

- Heterogeneous Graph Transformer (HGT): we basically built this HGNN based on the HGTConv operator available in Pytorch. In our implementation, for each node type, we first pass the input to a Linear layer, followed by a ReLU activation function. Then, the output is passed to 2 HGT convolutional layers (the input and output sizes of these hidden channels were set equal to 64). A final Linear layer was applied to each node type.
- Heterogeneous Attention Networks (HAN): the structure of this HGNN is identical to the previous one, with the difference that the applied internal convolution operators were HANConv.
- Heterogeneous SAGE: In this case, a HeteroConv wrapper has been used to wrap different SAGE convolutional operators. SAGEConv is a very powerful operator that learns node representations by aggregating information from the neighborhood of each node [8]. Here, for each type of edge contained in the dataset, a different SAGEConv operator has been deployed, to capture all the possible relationships between nodes. A ReLU activation function was used at the end of each layer. Finally, a different Linear layer was added, on for each node type for the final classification, i.e., in this case two Linear layers, one for the transactions and one for the wallets.
- Heterogeneous GAT: In this case, the HeteroConv operator is used to wrap different GAT convolutional layers, similarly to what has been done for Heterogeneous SAGE. Graph Attention network (GAT) is an operator which leverages self-attentional layers, enabling implicitly the capability, for each node, of giving weights to other nodes in a neighborhood [19].

3.3 Metrics

To evaluate the performance of the models, we used the following metrics, only for the illicit class: precision, recall, F1 score and micro-averaged F1 score.

- **Precision:** Precision measures the proportion of true positive predictions out of all positive predictions made by the model. A high precision score indicates that the model produces fewer false positives, meaning it accurately identifies illicit transactions (or wallets) when it labels a transaction (wallet) as illicit.
- **Recall:** it focuses instead on false negative errors, giving us an indication of missed positive predictions, meaning illicit transactions (wallets) classified instead as licit. In other

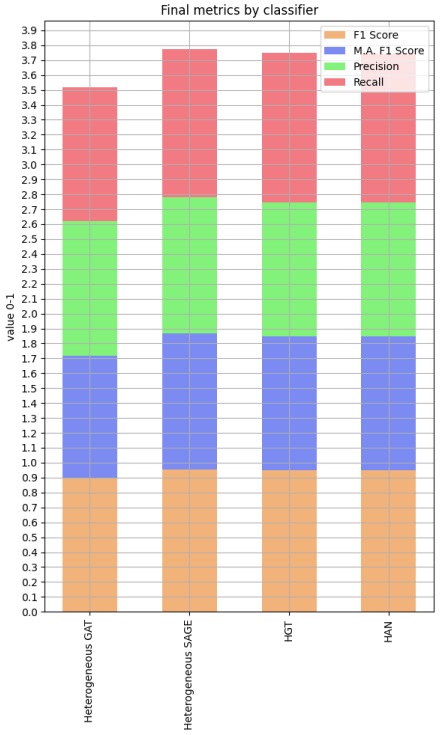

Figure 2: Classification results for the transactions.

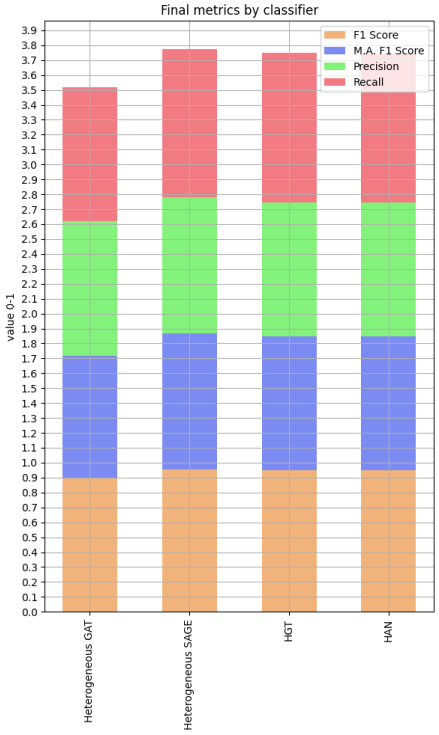

Figure 3: Classification results for the wallets.

words, it evaluates how well the model captures and identifies all illicit transactions (wallets).

- **F1 score:** it tries to find the balance between precision and recall by calculating their harmonic mean. It thus provides a balanced evaluation of a model's performance.
- **Micro-avg F1 score:** The F1 Micro AVG calculates the F1-score at the micro-level by considering the total number of true positives, false positives, and false negatives across all classes. In other words, it indicates the proportion of correctly classified observations out of all observations (both licit and illicit).

Table 3: Metrics for transaction

Model	Precision	Recall	F1	F1 Micro AVG
Heter GAT	0.900	0.899	0.900	0.819
Heter SAGE	0.914	0.995	0.953	0.912
HGT	0.900	1.000	0.947	0.900
HAN	0.900	0.999	0.947	0.900

4 RESULTS

Table 3 shows the obtained results for the classification of transactions. Figure 2 shows these results in a stacked way, in order to better visualize the overall performance. The Heterogeneous GAT model achieves balanced precision and recall, resulting in a slightly weak F1 score. However, the F1 micro average indicates it has poorer overall performance, probably due to class imbalances. The Heterogeneous SAGE model performs better in recall, suggesting it effectively identifies positive instances. Its high F1 score indicates a good balance between precision and recall. The F1 micro average confirms its overall good performance. Indeed, in terms of F1 score and F1 micro average, this is the best classifier. The HGT model achieves perfect recall, capturing all positive instances. While a recall of 1 might seem ideal at first glance, it is important to consider that achieving a recall of 1 often comes at the expense of precision, which measures the proportion of identified positives that are actually correct. Indeed, in this case, the precision of HGT is slightly lower than Heterogeneous SAGE, indicating the presence of false positives. However, HGT has equal precision of the other remaining models. The F1 score balances these trade-offs, and the F1 micro average suggests good performance. The HAN model demonstrates high recall, minimizing false negatives. Its precision is also reasonable. The F1 score and F1 micro average validate its effectiveness across the dataset.

Table 4: Metrics for wallets

Model	Precision	Recall	F1	F1 Micro AVG
Heter GAT	0.922	1.000	0.959	0.921
Heter SAGE	0.948	0.897	0.922	0.860
HGT	0.975	0.820	0.891	0.815
HAN	0.922	0.999	0.959	0.921

Table 4 and Figure 3 show results similar to the previous ones, but applied to the classification of wallets. The Heterogeneous GAT model has a recall equal to 1, catching all positive instances. Again, at first glance, this result might indeed appear concerning. Upon closer examination, precision remains encouraging. The F1 micro average hints at potential challenges with class distribution. The Heterogeneous SAGE model has a very good value of precision. Its F1 score suggests a trade-off—some true positives sacrificed for fewer false positives. Thus, class imbalance might play an important role here. Similar but worse results are obtained for HGT. Finally, HAN provide a reasonable balance between precision and recall. In terms of Fi score and F1 micro average, this is the best classifier together with Heterogeneous GAT.

In conclusion, all these results suggest that representing the dataset as a heterogeneous graph and analyzing it using HGNNs, allows to perform a good classification. Thus, this approach might represent a viable strategy to identify token illicit misuses and then to improve the security of social systems based on tokenization.

5 CONCLUSIONS

In this paper, we have explored the possibility of using heterogeneous graphs and the potential of Heterogeneous Graph Neural Networks (HGNNs) in enhancing the security of complex social systems by identifying illicit and malicious behaviors. Our focus was on digital asset tokenization and the possibility to identify illicit activities. We focused on a dataset, Elliptic++, that allows to perform a supervised machine learning classification of transactions in Bitcoin. The results on the resulting heterogeneous graph show the viability of the approach and the importance of characterizing and describing interactions in these complex systems. Our idea is that not only these approaches allow for the identification of illicit token transactions, but they might be employed in broader contexts, for both studying system dynamics and activating mechanisms to cope with cybersecurity misuses and usurpation of resources in social systems.

There are several directions for future work. First, while the Elliptic++ dataset provides a valuable resource, the exploration of other datasets, particularly those representing different types of social systems, could provide additional insights and validate the generalizability of our approach. We plan to obtain other datasets related to other use-case applications, and apply the same analysis framework. The integration of the proposed approach into a real-world system, and the evaluation of its performance and impact, would be a significant step towards the practical application of these techniques.

As a further work, the development and evaluation of other types of HGNNs, or other graph-based machine learning models, could further improve the performance of the machine learning analyses.

REPRODUCIBILITY OF RESULTS

The results presented in this paper are based on the Elliptic++ that is available here: https://www.github.com/git-disl/EllipticPlusPlus. The full source code used for the metrics computation and the reported analysis is available here: https://github.com/stefano-ferretti/-heteroneneousGNN-AML/ with an open source license.

ACKNOWLEDGMENTS

This work is partially supported by the European Union - NextGenerationEU within the framework of PNRR Mission 4 - Component 2 - Investment 1.1 under the Italian Ministry of University and Research (MUR) programme "PRIN 2022" - grant number 2022N2NH42 SmartShires – CUP: H53D23003570006.

REFERENCES

[1] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74 (Jan 2002), 47–97. Issue 1. https://doi.org/10.1103/RevModPhys.74.47

[2] Saqib Ali, Guojun Wang, Komal Fatima, and Pin Liu. 2019. Semantic Knowledge Based Graph Model in Smart Cities. In *Smart City and Informatization*. Springer, 268–278. https://link.springer.com/chapter/10.1007/978-981-15-1301-5_22

[3] R. Bing, G. Yuan, M. Zhu, and et al. 2023. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artificial Intelligence Review* 56 (2023), 8003–8042. https://doi.org/10.1007/s10462-022-10375-2

[4] DeepAI. 2021. A Heterogeneous Graph Learning Model for Cyber-Attack Detection. *DeepAI* (2021). https://deepai.org/publication/a-heterogeneous-graph-learning-model-for-cyber-attack-detection

[5] Junting Duan, Yujie Luo, Zhicheng Zhang, and Jianjian Peng. 2024. A heterogeneous graph-based approach for cyber threat attribution using threat intelligence. In *Proceedings of the 2024 16th International Conference on Machine Learning and Computing* (Shenzhen, China) *(ICMLC '24)*. Association for Computing Machinery, New York, NY, USA, 87–93. https://doi.org/10.1145/3651671.3651707

[6] Youssef Elmougy and Ling Liu. 2023. Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 3979–3990. https://doi.org/10.1145/3580305.3599803

[7] Stefano Ferretti, Gabriele D'Angelo, and Vittorio Ghini. 2016. Smart multihoming in smart shires: Mobility and communication management for smart services in countrysides. In *2016 IEEE Symposium on Computers and Communication (ISCC)*. 970–975. https://doi.org/10.1109/ISCC.2016.7543862

[8] Will Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems (NeurIPS)* (2017).

[9] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2704–2710. https://doi.org/10.1145/3366423.3380027

[10] Zhigang Jin, Xiaofang Zhao, and Yuhong Liu. 2021. Heterogeneous Graph Network Embedding for Sentiment Analysis on Social Media. *Cognitive Computation* 13 (2021), 81–95.

[11] Anelia Kurteva and Anna Fensel. 2021. Enabling Interpretability in Smart Cities with Knowledge Graphs: Towards a Better Modelling of Consent. *IEEE Smart Cities* (2021). https://smartcities.ieee.org/newsletter/june-2021/enabling-interpretability-in-smart-cities-with-knowledge-graphs-towards-a-better-modelling-of-consent

[12] Suhui Liu, Jiguo Yu, and Liquan Chen. 2021. Rewarding and Efficient Data Sharing in EHR System with Coalition Blockchain Assistance. In *Wireless Algorithms, Systems, and Applications*. Springer, 95–107.

[13] Yuke Ma, Yonggang Chen, Yanjun Wang, Jun Yu, Yanting Li, Jinyu Lu, and Yong Wang. 2023. The Advancement of Knowledge Graphs in Cybersecurity: A Comprehensive Overview. In *Computational and Experimental Simulations in Engineering*. Springer, 65–103. https://link.springer.com/content/pdf/10.1007/978-3-031-42987-3_6.pdf?pdf=inline%20link

[14] Negar Maleki, Balaji Padmanabhan, and Kaushik Dutta. 2022. Representing Social Networks as Dynamic Heterogeneous Graphs. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. 1–10. https://doi.org/10.1109/ICDMW58026.2022.00098

[15] S. Marasi and S. Ferretti. 2024. Anti-Money Laundering in Cryptocurrencies Through Graph Neural Networks: A Comparative Study. In *Proceedings of the IEEE Consumer Communications & Networking Conference (CCNC 2024)* (Las Vegas, USA). IEEE ComSoc.

[16] Nicola Pocher, Marco Zichichi, Fabio Merizzi, Muhammad Zeeshan Shafiq, and Stefano Ferretti. 2023. Detecting Anomalous Cryptocurrency Transactions: an AML/CFT Application of Machine Learning-based Forensics. *Electronic Markets* 30 (2023). https://doi.org/10.1007/s12525-023-00654-3

[17] PyTorch Geometric Team. 2024. HeteroConv: A Generic Convolutional Wrapper for Heterogeneous Graphs. https://github.com/pyg-team/pytorch_geometric. Accessed: yyyy-mm-dd.

[18] Lan van Wassenaer, Cor Verdouw, Ayalew Kassahun, Burak van Hilten, Mireille, Koos van der Meij, and Bedir Tekinerdogan. 2023. Tokenizing circularity in

agri-food systems: A conceptual framework and exploratory study9. *Journal of Cleaner Production* 413 (2023).

[19] Petar Velivckovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

[20] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2022–2032. https://doi.org/10.1145/3308558.3313562

[21] Qiang Yang, Qiannan Zhang, Chuxu Zhang, and Xiangliang Zhang. 2022. Interpretable Relation Learning on Heterogeneous Graphs. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1266–1274. https://doi.org/10.1145/3488560.3498508

[22] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 793–803. https://doi.org/10.1145/3292500.3330961

[23] Shuai Zhang, Zhou Zhou, Da Li, Youbing Zhong, Qingyun Liu, Wei Yang, and Shu Li. 2021. Attributed Heterogeneous Graph Neural Network for Malicious Domain Detection. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 397–403. https://doi.org/10.1109/CSCWD49262.2021.9437852

[24] Xingchen Zou, Jiani Huang, Xixuan Hao, Yuhao Yang, Haomin Wen, Yibo Yan, Chao Huang, and Yuxuan Liang. 2024. Learning Geospatial Region Embedding with Heterogeneous Graph. *arXiv preprint arXiv:2405.14135* (2024). https://arxiv.org/abs/2405.14135