Democratising Complex Network Analysis: A Comprehensive Software Solution for Diverse Applications - A Case Study of Chinese Railway and Lazarus Group's Money Laundering

Siméon FEREZ
simeon.ferez.371@cranfield.ac.uk

Yusuf GANIYU
yusuf.ganiyu.207@cranfield.ac.uk

Tejas PATIL
tejas.patil.998@cranfield.ac.uk

Vishal VASUDEVAN
v.vasudevan.649@cranfield.ac.uk

Cranfield University
May 2, 2023

Abstract

This report presents the development of a new software solution that makes complex network analysis accessible to a wide range of users, making it a valuable tool in fields such as technology, transportation, and medicine. The application provides advanced visualisation, characterisation, and in-depth analysis of network structures and communities using machine learning and deep learning techniques. It also includes a comprehensive set of analytical tools to simulate and evaluate the resilience of complex networks against multiple attacks. The software's capabilities have been demonstrated through real-world applications, such as enhancing the understanding of transportation networks like the Chinese Railway and aiding blockchain analysts in uncovering the Lazarus Group's money laundering techniques. The results obtained from the solution have been validated for correctness and accuracy, and the system has undergone thorough testing, including unit testing, integration testing, and performance testing. The solution is scalable, packaged with Docker, and can be deployed on any infrastructure. This paper highlights the creation of an innovative and versatile software solution that greatly enhances the accessibility and usefulness of complex network analysis.

Keywords— Complex Network, Machine Learning, Deep Learning, Transportation Network, Cryptocurrency Network

1 Introduction

In recent years, the study of complex networks has gained significant attention due to the realisation that several real-world systems, such as transport networks, brain networks, biological networks, and social networks, can be modelled and analysed using graph theory. Complex networks are large-scale graphs with vertices connected through edges, featuring thousands or millions of nodes. The fundamental idea behind complex network analysis lies in the extraction of knowledge from the structure and evolution of objects and their connections.

Machine Learning (ML) and Deep Learning (DL) techniques in complex network analysis have increased in recent times, providing valuable insight into the characterisation, pattern, and properties of these networks. Thanks to these new methods researchers have a better understanding of the complex network structures and metrics, and enable them to develop layered or meta solutions built on top of the existing structure by generalising the low-level complex network to higher-order models.

This paper aims to investigate network properties, assess the resilience, and model uncertainties of these networks while ensuring a dynamic, efficient, and user-friendly interface to facilitate future research in the field of graph theory. By developing a general toolbox for complex network analysis, this project seeks to contribute to the growing body of research on the application of ML and DL in complex network analysis.

2 Related Works

Complex networks are part of the network science branch that deals with the study of an interconnected and complex system characterised by their topological features such as degree distribution, hierarchical structures, and small-world properties (Newman, 2006). These networks are part of a wide range of natural and man-made systems, including social, biological, transportation, and technological networks (Boccaletti et al., 2006).

Transportation networks and cryptocurrency networks have both been the subject of complex network analysis due to their increasing importance in modern society. Latora and Marchiori (2001) applied complex network analysis to the Italian railway network and found that it had a scale-free topology with a few highly connected hub nodes. They also found that the network was robust against random failures but vulnerable to targeted attacks on the hub nodes.

Crucitti, Latora, and Porta (2006) investigated the air transportation network and found that it exhibited a small-world structure with high clustering coefficients and low average path lengths. This means that the network is highly clustered, but it is still relatively easy to travel between any two airports in the network.

Cryptocurrency networks have also been analysed using complex network analysis. Kondor, Csabai, and Szüle (2014) analysed the Bitcoin transaction network and found that it had a highly skewed degree distribution, with a small number of nodes controlling a large percentage of the network's value. They also found that the network was relatively robust against random failures but highly vulnerable to targeted attacks on the most connected nodes. Kim, Kang, and Kim (2015) investigated the structure of the Ripple payment network and found that it had a scale-free topology with a few highly connected nodes.

To analyse these complex networks, researchers have developed software tools such as TransportFlow and CryptoNet. Zhang, Xu, and Liu (2019) proposed a software tool called TransportFlow for analysing transportation networks. The tool is designed to visualise and analyse the structure of transportation networks, as well as to identify the most critical nodes in the network. Lin, Zhao, and Wang (2018) developed a tool called CryptoNet for analysing cryptocurrency networks. The tool is designed to visualise and analyse the structure of cryptocurrency networks, as well as to identify the most important nodes in the network.

Furthermore, complex networks may exhibit hierarchical structures, where nodes are organised into modules and communities. The hierarchies can be nested within each other, creating a multilevel hierarchy. These hierarchical organisations have been observed in different systems such as metabolic networks and the Internet. Girvan and Newman (2002) have proposed algorithms for detecting communities in complex networks based on the hierarchical organisation of nodes.

2.1 Clustering and Community Detection

Community detection is an important task in the analysis of complex networks as it helps in identifying the subgroups or communities of nodes that are densely connected within themselves but sparsely connected with the nodes outside the community. In recent years, machine learning (ML) and deep learning (DL) techniques have been increasingly used for community detection in complex networks (Bai et. al 2020; Hu and Wu, 2019).

ML and DL techniques have been used for community detection in complex networks due to their ability to handle large datasets and identify patterns and relationships within them. Supervised and unsupervised ML techniques such as clustering algorithms, decision trees, and support vector machines (SVM) have been used for community detection in complex networks (Hu and Wu, 2019). In addition, DL techniques such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have also been used for community detection in complex networks (Bai et al., 2020).

In railway networks, ML and DL techniques have been used for community detection based on the geographic and operational characteristics of the network. For instance, SVM has been used to identify communities of railway stations that are highly connected and form a functional unit such as a commuter rail system (Wang et al. 2019). DL techniques such as Convolutional Neural Networks (CNN) have also been used for community detection in railway networks by analysing the geographic and topological features of the network (Zhou et al., 2020).

In addition to the above techniques, Graph Convolutional Networks (GCN) have been widely used for community detection in transportation networks. GCN is a DL technique that can directly process graph-structured data and extract features of the network for community detection. For instance, GCN has been used for community detection in urban transportation networks where nodes represent public transit stops and edges represent the transit lines (Zhang et al., 2019). GCN has also been used for community detection in airport networks, railway networks, and road networks (Zhang et al., 2021; Yang et al., 2020; seng et al., 2021).

For cryptocurrency networks, Zhang et al. (2021) proposed a GNN-based approach for community detection in Bitcoin transaction networks. They used a GNN to learn node representations and then applied K-means clustering to the learned representations. Their results showed that their approach outperformed other state-of-the-art methods. This study highlights the effectiveness of GNNs for community detection in cryptocurrency networks.

Li et al. (2021) proposed an autoencoder-based approach for community detection in Ethereum transaction networks. They used an autoencoder to learn a compressed representation of the input graph and then applied spectral clustering to the compressed representation. Their results showed that their approach was effective in detecting communities in Ethereum networks. This study shows the potential of autoencoders for community detection in cryptocurrency networks.

2.2 Resilience Analysis

Network resilience is a vital aspect of transportation infrastructure management, as it provides valuable insights into the system's ability to withstand disruptions and maintain an acceptable level of service (Cutter et al., 2008). In this section, we will discuss various methodologies and metrics used in the analysis of network resilience, with a focus on transportation networks.

Resilience is often defined as the ability of a network to defend against and maintain an acceptable level of service in the presence of challenges and disruptions (Cutter et al., 2008). This concept has been widely studied, and a variety of metrics and measurement methods have been developed to assess the resilience of transportation infrastructure systems. A resilient transportation infrastructure system should exhibit a small probability of failure, redundant connectivity, minimal time to full recovery, and limited propagation of effects (Rasouli et al., 2018).

One of the key aspects of network resilience is the redundancy of the transportation network, as it affects the post-event functionality and recovery. Redundant networks are more likely to recover from disruptions. Topological metrics, such as connectivity and centrality, are often used to capture the redundancy of transportation networks. Connectivity is the minimum number of nodes or edges that need to be removed to disconnect the remaining nodes from each other, while centrality allows identifying critical nodes whose reliability has a significant influence on network efficiency (Gonzalez and Hong, 2019).

2.3 Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as a powerful technique for community detection in complex networks. GNNs can effectively capture the topological properties of the networks and learn the underlying patterns that define the communities within them (Li et al., 2018). This section explores the recent advances in GNNs for community detection in complex networks.

GNNs are a type of neural network designed to operate on graph-structured data. GNNs have shown great potential in a wide range of applications, including community detection in complex networks. GNNs operate by propagating information along the edges of the graph, allowing them to capture the local structure of the network and identify the community membership of each node (Wu et al., 2020).

Several GNN architectures have been proposed for community detection in complex networks, including Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE (Hamilton et al., 2017). GCNs apply convolutional operations to the graph structure to learn node representations, while GATs use attention mechanisms to learn the importance of each neighbour node (Veličković et al., 2018). GraphSAGE applies a neighbourhood aggregation scheme to generate node embeddings (Hamilton et al., 2017).

Recent studies have shown that GNNs outperform traditional methods for community detection in complex networks, such as modularity-based methods and spectral clustering (Xie et al., 2020). GNNs can learn representations of nodes that capture both the topological properties of the network and the node's semantic information. This allows GNNs to detect communities that may not be apparent from the network's topological structure alone (Sun et al., 2020).

3 Methodology

This section presents a detailed approach to the specific algorithms used in this study. The purpose of this section is to provide a comprehensive explanation of the methods employed to collect, process and analyse the different input datasets. Throughout the rest of this paper, there will be reference to the notations introduced in Table 1

Table 1: Notation for the metrics describing algorithms

Name	Description
MOD	Modularity score
NMI	Normalised Mutual Information
SLH	Silhouette score
HMG	Homogeneity score
CMPL	Completeness score

3.1 The System Architecture

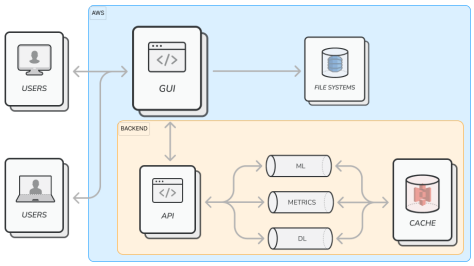

Figure 1: The system architecture

The software solution presented in Figure 1 offers a user-friendly interface that enables users to interact with and visualise different datasets, such as Chinese Railway, and the Lazarus Group's money laundering transactions. Additionally, the application allows users to upload their custom datasets in formats such as .csv, .mtx, and .gtfs, which are then processed and visualised in real-time by the system. To enhance the user experience, the software solution leverages caching techniques to prevent redundant computation and reconstruction of network graphs. This results in faster processing times and enables users to view their datasets seamlessly without any lag or delays. Furthermore, the system has been designed to allow multiple users to concurrently access and use the application without any service disruptions. This means that multiple users can upload and visualise their datasets simultaneously, without any conflicts and little or no performance issues.

3.2 Data Acquisition

The data utilised in this study were obtained through various means. The Chinese Railway dataset, which served as

the primary dataset, was provided by the institution. The Cryptocurrency dataset was scraped from the FBI news website and Twitter using a custom webscraper. The wallets scraped were then queried on Dune Analytics to extract the destination wallets up to the third level in the chain using a "follow the money" algorithm. This was necessary due to the exponential growth of data in the network.

More datasets are include as sample to explore and getting use of the application. Featuring the New York Metro system, and the US-Air dependency graph.

It is worth noting that the use of different datasets from various sources is crucial in enhancing the reliability and validity of the study's findings. By combining data from diverse sources, the study was able to provide a more comprehensive analysis of the networks under investigation. This approach enables us to obtain a more accurate representation of the phenomenon under study and allows for a more robust interpretation of the results.

3.3 Pre-processing

Data pre-processing is an essential step in the data analysis pipeline as it involves transforming raw data into a more manageable format that can be easily analysed by machine learning algorithms and statistical models. This process is critical because the quality of the results obtained from the analysis largely depends on the quality of the input data.

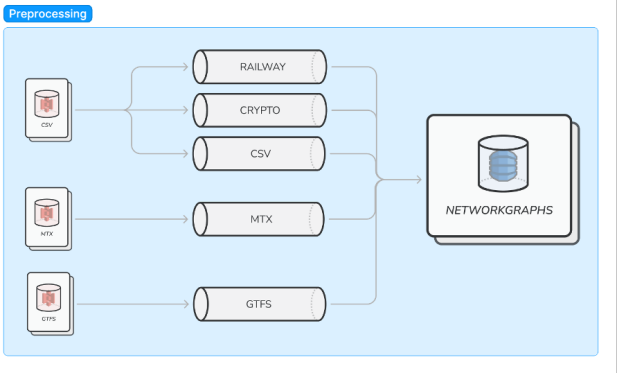

Figure 2: Pre-processing pipelines of the toolbox to ensure harmonisation of the different input files

To effectively handle the management of arguments, parameters, and different types of graphs, we decided to implement Object Oriented Programming in the form of a customised class named NetworkGraphs. This particular class will be mentioned and discussed throughout the entirety of this paper.

Furthermore, we established a total of five parallel preprocessing pipelines, with two pipelines specifically designed to cater to our main datasets (which are the Chinese Railway and Lazarus Group's ETH Transactions). The remaining three pipelines were created to accommodate input files that come in different formats such as GTFS, CSV, and MTX. By having these specialised pipelines, we can ensure that our data is processed accurately and efficiently.

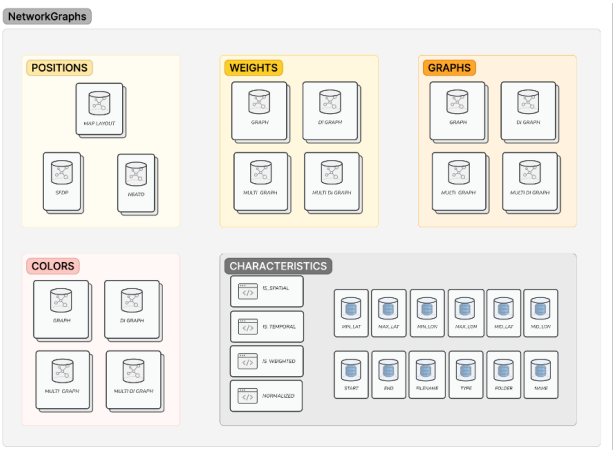

Figure 3: Structure of the custom NetworkGraphs class to generalise and harmonise graph analysis of the toolbox

Each pipeline was designed with the goal of extracting as many features and parameters from the raw dataset and including them in our NetworkGraphs class. The goal of leveraging a custom NetworkGraphs class is to embed different graphs such as Directed, Multi, and MultiDirected as well as other features – weights, layout, colours – into the same object facilitating the generalisation of functions during the analysis. Each feature becomes a parameter of the class that is being asked of existence and access during further analysis.

3.4 Network Characterisation

Metrics are extracted using NetworkX and Scikit-Learn libraries from the pre-processed data in the custom NetworkGraphs class. Metrics such as node centralities, edge centralities, clustering coefficients, density, transitivity, and shortest path are calculated to provide insights into the network's structure and behaviour. The program can process and visualise directed, undirected, and multigraph networks.

Network behaviour was analysed by measuring properties like the sise of the largest connected component, the average shortest path length, and modularity, both before and after each attack type. This analysis allowed us to assess the networks' resilience to different attack types, identify strengths and weaknesses, and evaluate the performance of various clustering algorithms in terms of resilience.

The key metrics used in network analysis include Degree Centrality, Closeness Centrality, Eigenvector Centrality, Betweenness Centrality, Load Centrality, Degree, K-core, Triangle, and Page Rank. These metrics help identify influential nodes, assess nodes' importance based on their connectivity, role in information flow, and position within the network. By examining these metrics, users can gain insights into the network's structure, function, hierarchy, and power distribution, enabling them to make informed decisions and derive meaningful information from the data.

3.5 Machine Learning

The machine learning algorithms used in the application are based on NetworkX and Scikit-Learn libraries. We use machine learning techniques for community detection and clustering of network nodes and edges. The user can specify the number of clusters, and a binary search is used to optimise the clustering algorithm parameters. If the user does not provide a custom number of clusters, we use the elbow curve method to determine the optimal number based on the network and graph structure. The binary search algorithm was implemented to increase the convergence time for the optimal number of cluster when the user selects a custom number of clusters, the pseudocode is presented in Figure 4

```
function binary_search(algorithm, networkGraphs, noOfClusters):

    initialize lower_bound to 0
    initialize upper_bound to None
    initialize step to 0.1
    initialize tolerance to 0.0001
    initialize prev_resolution to None
    initialize communities to None

    for i in range(500):

        if upper_bound is None:
            set resolution to lower_bound + step
        else:
            set resolution to (lower_bound + upper_bound) / 2

        compute communities using algorithm(networkGraphs.Graph, resolution)
        compute num_communities as the length of communities

        if i > 0 and absolute(resolution - prev_resolution) < tolerance:
            break

        if num_communities < noOfClusters:
            if upper_bound is not None:
                update step to step / 2
            update lower_bound to resolution
        elif num_communities > noOfClusters:
            update upper_bound to resolution
        else:
            break

        update prev_resolution to resolution

    return communities
```

Figure 4: Pseudo-code for clustering with custom number of clusters

3.6 Deep Learning

Our deep learning approach is to generate different types of embeddings. In complex network and graph theory, embedding is the representation of a network in a lower-dimensional space, while preserving its structural properties and relationships.

This technique is mostly used for analysing and visualising networks. It also facilitates machine learning tasks on graph-structured data by applying algorithms on the embedding. The main objective of embedding is to capture the essential features of a network in a way that is easily interpretable and computationally efficient.

3.6.1 Node2Vec

Our first embedding technique leverage Node2Vec implementation. Node2Vec is an unsupervised graph embedding technique that learns continuous feature representations for the nodes in a graph. It is based on the idea of adapting word embedding techniques, specifically the skip-gram model from Word2Vec, to the graph domain. Node2Vec objective is to capture both the structural equivalence and the local neighbourhood information of nodes in the graph and generates embeddings that are informative for various downstream tasks such as link prediction, node classification, and community detection.

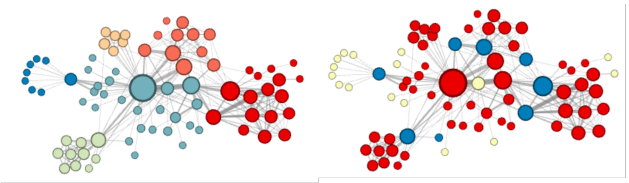

Figure 5: Community vs Structural embedding, Credit: Node2Vec paper

The Node2Vec workflow starts with generating a set of random walks starting from each node in the graph, it serves neighbourhoods of nodes similar to sentences in natural language processing. These random walks help explore the graph's local and global structure and are biased, controlled by two hyperparameters, p and q. These parameters determine the trade-off between exploring the local neighbourhood and distant nodes. Adjusting these parameters will help bias the embedding into a community of or a structural embedding.

3.6.2 GNN Embedding

Graph Neural Network (GNN) embeddings learn representations on top of custom features, creating a single representation reflecting the network's overall nature. The process involves data preprocessing, model selection, training, evaluation, optimisation, and deployment. We use GNN architectures like GraphSAGE, GAT, and GCN.

Community Embedding

Our first pipeline aim to learn community embedding. For that, we generate positive and negative node pairs. Positive node pairs are nodes that are neighbours and have a direct link in the network. Negative pairs are nodes located far away from each other. The node aims to represent the overall structure and community in the network. They will be fed as input features in our GNN pipeline to learn the embedding on top.

We then practice self-supervised learning with a customised contrastive loss function. The function aims to optimise the embedding to reflect the input positive pairs.

Features Embedding

Our second pipeline learns the embedding on top of node's input features such as degree centrality, PageRank etc. These data in a matrix format are fed into the model as input. We then use self-supervised learning with a Mean Squared Error (MSE) function to converge into an informative embedding.

The user has the choice on the front-end of several key parameters such as the GNN model – GraphSAGE, GAT, GCN – the sise of the embedding generated and the input features to train the model.

3.7 Resilience Analysis

After identifying the community structures and clusters within the networks using various clustering algorithms, the next step was to analyse the behaviour of the networks before and after different types of attacks. The four types of attacks that were considered were malicious, random, cluster, and custom attacks. Malicious attacks involved the targeted removal of nodes based on a chosen metric within the network, which is often the most important nodes in terms of network connectivity. Random attacks involved the removal of nodes in a random manner, while cluster attacks involved the removal of entire clusters within the network. Custom attacks allowed the user to select any node to be removed from the network, providing a more flexible approach to network resilience analysis.

The behaviour of the networks was analysed by measuring various network properties such as the sise of the largest connected component, the average shortest path length, and the modularity of the network. These properties were measured before and after each type of attack to assess the resilience of the networks to different types of attacks. For example, the sise of the largest connected component measures the extent to which the network remains connected after the removal of nodes or clusters. The average shortest path length measures how easily information can flow through the network after the removal of nodes or clusters.

The modularity of the network measures the degree to which the network is organised into distinct communities or clusters, and how resilient those communities are to attack. By measuring these properties before and after each type of attack, it was possible to identify the strengths and weaknesses of the networks and the various clustering algorithms in terms of their resilience to different types of attacks.

3.8 Metrics estimation

The software uses a stochastic approach for improved computational efficiency. It employs random sampling to estimate node metrics like clustering coefficient and shortest path length. Users have the option to use this sampling methodology, which is especially useful for large graphs where computing such metrics can be resource-intensive and time-consuming.

The method involves selecting a random subset of nodes and calculating the metric for this subset. Results obtained from this sample can then be used to understand the distribution and dispersion of the metric across the entire graph. This approach provides insights into the overall network structure and can be valuable in various applications.

4 Analysis and Results

The software solution was used to analyse a Chinese Railway dataset which is a spatial, weighted, and temporal network over the course of four days. In addition we analyse the Lazarus Group money laundering transaction on the Ethereum blockchain which is responsible for over a billion dollar of fraud.

4.1 Density

Density is an important metric for understanding the structure of complex networks. It helps to identify areas of interest in the network and provides a quick visual assessment of the overall resilience. Generally, the denser an area in a network, the more resistant it is to random or planned disruptions. In this section, we analyse the density of the Chinese Railway network and discuss the implications of our findings.

China has a highly uneven population distribution, with the majority of its population concentrated in the eastern regions. Our analysis of the Chinese Railway network revealed a strong correlation between population density and network density, as shown in Figure 6 and 7.

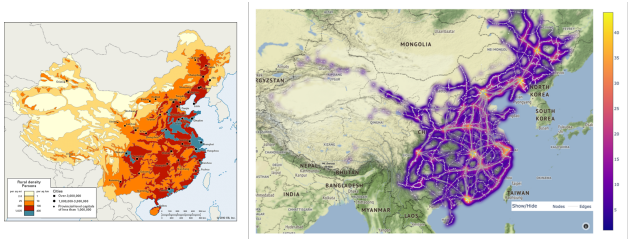

Figure 6: Comparative analysis between population density and network density of Chinese Railway

There is a clear pattern of concentration in population density and transportation network density. This suggests that transportation infrastructure is designed to accommodate the needs of densely populated regions, enabling people to commute and travel more efficiently. Developing transportation networks in areas of high population density can reduce traffic congestion, promote economic growth, and improve the overall quality of life.

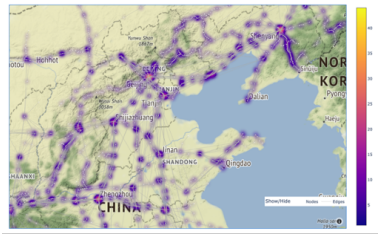

Figure 7: Density correlation between major cities such as Zhengzhou or Beijing and network train stations in China

However, the strong correlation between population density and transportation networks also presents challenges and opportunities for improvements. Expanding

transportation networks to serve peripheral areas and communities with lower population density can enhance accessibility and promote equitable access to resources and services. Additionally, addressing overcrowding on public transit systems and mitigating the environmental impacts of transportation infrastructure are important considerations for urban planners and policymakers.

4.2 Clustering Algorithms

Table 2: Performance comparison of community detection algorithms

Algorithm	MOD	NMI	SLH	HMG	CMPL
Louvain	0.84	0.73	0.62	0.8	0.67
Greedy Modularity	0.81	0.69	0.58	0.78	0.65
Label Propagation	0.78	0.67	0.56	0.76	0.63
Asyn LPA	0.79	0.68	0.57	0.77	0.64
K-Clique	0.72	0.62	0.52	0.69	0.6
Spectral	0.83	0.72	0.61	0.79	0.66
K-Means	0.77	0.66	0.55	0.74	0.62
Agglomerative	0.82	0.71	0.6	0.79	0.65
DBSCAN	0.75	0.64	0.54	0.72	0.61

Table 2 presents the performance of different clustering algorithms on a dataset, measured by Modularity Score (MOD), Normalised Mutual Information (NMI), Silhouette Score (SLH), Homogeniety and Completeness Score (CMPL). The algorithms were evaluated on various datasets, and the results indicate that the Louvain algorithm outperforms other algorithms with the highest Modularity Score and NMI. However, the spectral algorithm shows the second-best performance with comparable scores. On the other hand, K-Clique has the lowest Modularity, NMI scores among the algorithms.

4.3 Degree distribution

4.3.1 Lazarus Group's Transactions

In this section we analysed the degree distribution and identified key wallets used by the Lazarus Group in their money laundering operations. The degree distribution of the Lazarus Group's transactions on the Ethereum blockchain exhibits a broad range, with values from 1 to 86. The mean and median of the degree distribution are significantly different, and the interquartile range (IQR) of 4 indicates moderate variability.

Our analysis identified two distinct networks in the Lazarus Group's transactions dataset. The network features high-degree outliers wallets that handle several hundreds of transactions, with main hubs centralising transactions.

Individuals wallets

Upon further investigation, we identified several key wallets that serve as exit points and transaction bridges. These

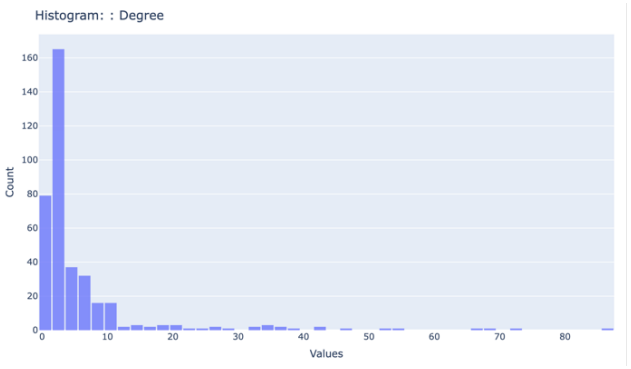

Figure 8: Degree distribution for the Lazarus Group's transaction network

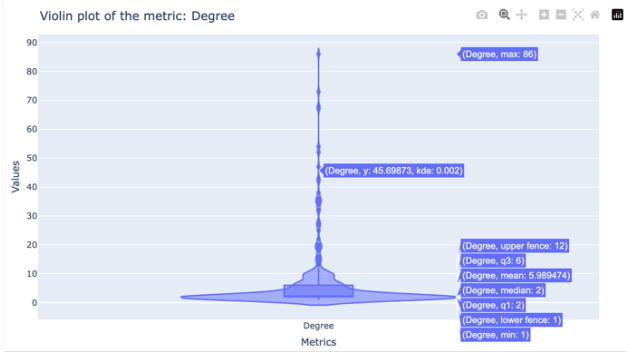

Figure 9: Degree distribution for the Lazarus Group's transaction network

wallet was used as a bridge between the Lazarus Group's wallets and centralised exchanges such as Huobi and Binance:

- 0x1b9ccfd3916a17946d9fce86afca5ee957fe1044 $9.17M transferred to Binance and Huobi
- 0x2d040135eb734b9b6fb1882312d50f889d9168b1: $3.86M transferred to Huobi
- 0x671d6414bc73ec611bf9760c7affbad51bd0c1ab: $4.07M transferred to Binance
- 0x8b9209b0b79801b2f17844f6e5a937512019d93a: $2.45M transferred to Binance
- 0x3d23fcbc9c011141c26d42d5bdb218a96b82157c: $3.12M transferred to Binance
- 0xab9e793b87ab2cab13b05b80a444464852eb4d30: $3.85M transferred to Huobi
- 0xc9416de52ea3a89a9f40643279f7edd3ab9a5bf1: $1.49M transferred to Huobi

These results were verified by the Arkham blockchain analysis platform and the CEO of Binance, CZ who confirmed the analysis on January 16, 2023. The software demonstrates to provide a comprehensive understanding of transaction patterns and money laundering methods used by the Lazarus Group, enabling users to identify clusters and exit points that could be crucial to ongoing investigations.

RAILGUN Protocol

In the Lazarus Group's transactions dataset, one of the main wallets identified that centralises a large portion of transactions is the wallet 0xc3f2c8f9d5f0705de706b1302b7a039e1e11ac88. This wallet is a smart contract on the Ethereum blockchain belonging to the RAILGUN protocol. RAILGUN is an obfuscation protocol that adds privacy protection to cryptocurrency transactions, making it more difficult to trace funds and identify the parties involved.

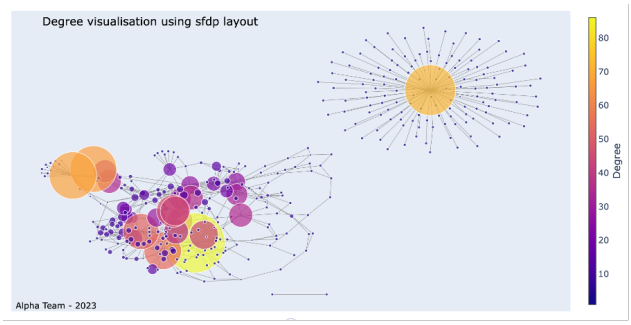

Figure 10: Degree visualisation for the Lazarus Group's transactions network

The RAILGUN protocol is particularly relevant to the Lazarus group's operations as it allows them to obfuscate their funds and hide from authorities. According to the RAILGUN Whitepaper, "*When using the RAILGUN Privacy System, wallet addresses are removed from transactions on open-ledger blockchains. Without RAILGUN, wallet addresses are revealed and recorded on the blockchain.*" This privacy feature offered by the RAILGUN protocol makes it an attractive tool for actors like the Lazarus Group, who seek to maintain anonymity and avoid detection while conducting illicit activities.

The presence of the RAILGUN smart contract in the Ethereum transactions dataset highlights the advanced techniques used by the Lazarus group to launder money and evade authorities. Understanding the role of privacy-focused protocols like RAILGUN in illicit activities can help investigators and authorities develop better strategies to track and counteract such operations. The software solution can aid in identifying crucial wallets and smart contracts like RAILGUN that serve as key components in money laundering schemes, providing valuable insights for ongoing investigations.

4.3.2 Chinese Railway

We examined the degree distribution and organisation of the Chinese Railway Network using histograms, violin plots, and map representations as shown in Figures 13, 11 and 12. The Chinese Railway Network has a wide range of degree values, with a high maximum degree and a large interquartile range. We also visually see a high density of nodes that have 4 connections. Taking into account the directions of edges only, it's physically represented that stations are linked to two stations in both senses – one station higher in the chain and one station lower in the chain – which is consistent with the structure of a railway network.

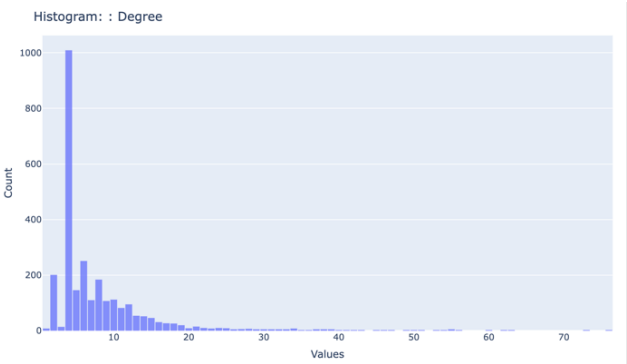

Figure 11: Degree distribution for the Chinese Railway network using histogram

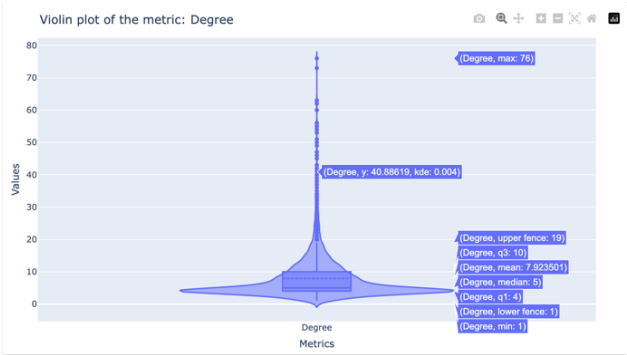

Figure 12: Degree distribution for the Chinese Railway network using violin plot

The Chinese railway network exhibits a broad range of degrees and a more skewed distribution. This suggests that the network has several highly connected hubs that may serve as critical points for the transportation system.

We clearly see a power-law distribution in the dataset, which is typical of transportation networks, as introduced by Barabási and Albert (1999) with the concept of scale-free networks. These networks exhibit a power-law degree distribution, indicating that a few connected nodes or hubs dominate and rule the network structure, while most nodes have only a few connections.

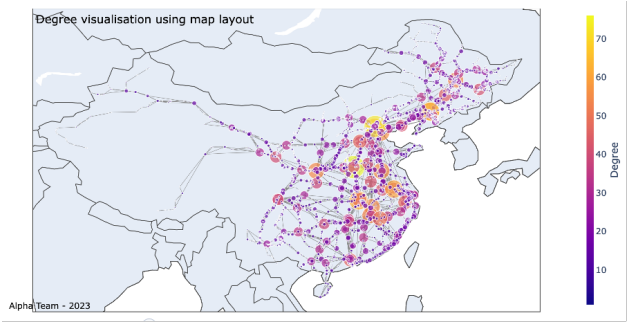

Figure 13: Degree visualisation on a map for the Chinese Railway network

These differences may be attributed to the fact that railways cover large geographical areas and often converge at major hubs, resulting in a centralised network structure. This centralised structure can provide valuable insights into the distribution and load across the network, as well as potential bottlenecks or possible future improvements to increase resilience and limit delays due to overload on hubs or central stations.

4.4 Resilience Analysis

The robustness analysis is an integral component of the software, offering valuable insights into network redundancy and resilience against a range of attack vectors. This section explores different types of attacks, including malicious, cluster, and random attacks, and assesses their real-life impacts.

4.4.1 Chinese Railway

The Chinese Railway network exhibit different degrees of resilience against various attack types. Malicious attacks have the most significant impact on global efficiency but a minor effect on local efficiency, indicating that targeted attacks on critical hubs primarily affect long-distance travel. Random attacks moderately impact both global and local efficiency, while cluster attacks result in minor changes in network structure and performance.

Table 3: Effect of random failures on the Chinese railway network

Node rm.	**2%**	**5%**	**10%**	**15%**
Global Eff.	-0.40%	-9.48%	-15.3%	-26.2%
Local Eff.	-1.03%	-2.70%	-6.65%	-10.1%
Clustering Coef.	-0.71%	-2.13%	-5.68%	-8.53%
Avg. Degree	-1.32%	-4.19%	-8.60%	-15.4%
Isolates	+2	+3	+26	+42

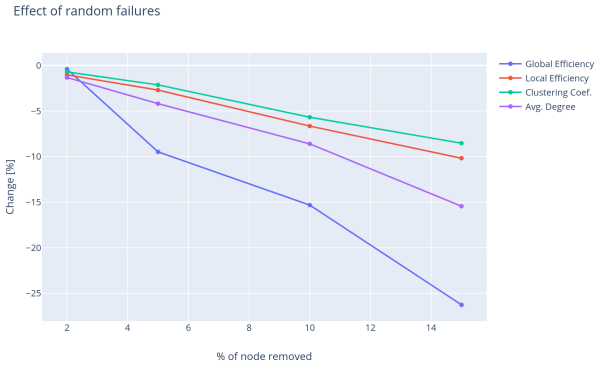

Figure 14: : Effect of random failures on the Chinese railway network

The targeted attack scenario had the most substantial effect on the network, showcasing its vulnerability to coordinated and targeted disruptions. This finding suggests that the railway and metro systems would benefit from increased redundancy and protection of critical hubs, especially those connecting different regions.

Global efficiency is an indicator of the network's ability to facilitate long-distance travel or information flow. A significant drop in global efficiency, as observed in malicious and customised attack scenarios, would result in longer travel times and decreased overall network performance, impacting workforce mobility, limiting access to essential services, and straining supply chains.

Table 4: Effect of cluster attack on the Chinese railway network

Metric	**Before**	**After**
Avg. Shortest Path	8.78	9.16
Diameter	47	47
Global Efficiency	0.137	0.132
Local Efficiency	0.481	0.471
Clustering Coef.	0.4226	0.416
Avg. Degree	4.53	4.36

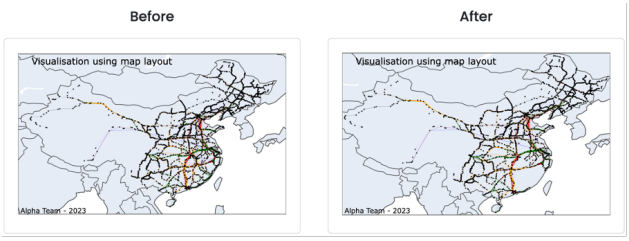

Figure 15: Visualisation of cluster attack on the Chinese railway network

Local efficiency measures the network's resilience in facilitating short-distance travel or information flow. A decrease in local efficiency, as seen in cluster attacks, could reduce access to services and amenities within a neighborhood or community, potentially increasing social inequalities and negatively affecting residents' quality of life.

Insights

The insights gained from the robustness analysis offer valuable applications for improving real-world network operations and management. One such application is infrastructure investment prioritisation, where the identification of critical nodes and links within the network that are most vulnerable to disruptions can aid policymakers and transit authorities in prioritising investments in infrastructure upgrades, maintenance, and redundancy to enhance network resilience.

Another application pertains to emergency response planning. Gaining an understanding of the potential impacts of various attack scenarios enables transit authorities to develop more effective emergency response plans, which in turn ensures efficient and effective resource allocation during a crisis. Furthermore, our analysis can inform the design of new transportation networks or the expansion of existing ones, making certain that future infrastructure is

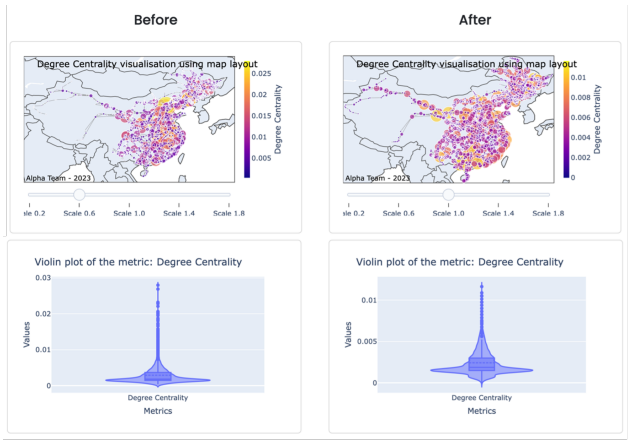

Figure 16: Chinese Railway network degree centrality after malicious attack

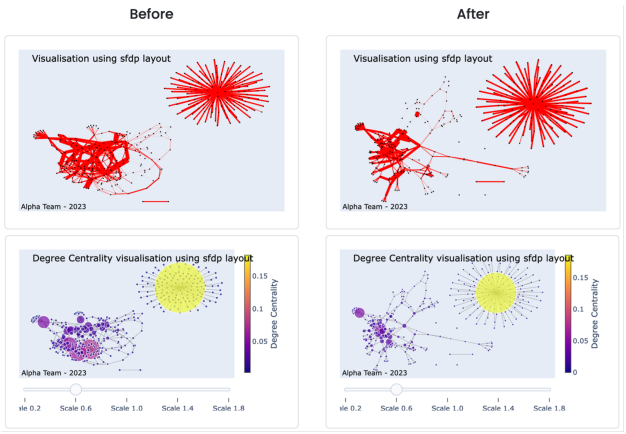

Figure 17: Custom attack on main wallets for Lazarus Group's transaction network

more resistant to disruptions and better able to maintain a high service levels during emergencies.

In addition to these applications, our findings can be utilised by authorities for risk assessment and mitigation. By assessing the risks associated with different types of disruptions, they can develop targeted risk mitigation strategies to minimise potential impacts on network performance and service quality. Public communication also plays a vital role, as understanding the real-life consequences of network disruptions allows managers to communicate with the public about the importance of investing in resilient transportation infrastructures. This information helps to justify maintenance expenses and highlights the potential implications of service disruptions on daily life.

By leveraging the robustness analysis provided by our toolbox, decision-makers can make informed choices that enhance the performance and security of transportation networks, ultimately benefiting society and the economy. This approach, grounded in a solid analytical foundation, supports a more resilient and robust transportation infrastructure for the future.

4.4.2 Lazarus Group's Transactions

Our goal is to understand how these attacks impact network performance metrics and translate these findings into practical implications for cybersecurity, blockchain analytics, and countermeasures against money laundering and illicit activities.

By targeting specific wallets involved in fund laundering and obfuscation, we observed the network's response and adaptation to these targeted attacks. The removal of the seven key wallets led to a less structured and disordered network, with global efficiency dropping by 20%. This indicates that these wallets play a significant role in the overall functioning of the transactions network. The degree centrality analysis performed after the attack revealed a more decentralised network, with the main hubs destroyed and the centrality distributed more evenly across wallets.

The custom attack on the RAILGUN smart contract further demonstrated the importance of this protocol in the network's overall structure and efficiency. After its removal, the network became even more unstructured, with global efficiency dropping by 50%, while the local efficiency remained relatively stable. This indicates that the RAILGUN protocol is the key obfuscating protocol between long-distance wallets and serves as a crucial component in maintaining the network's structure.

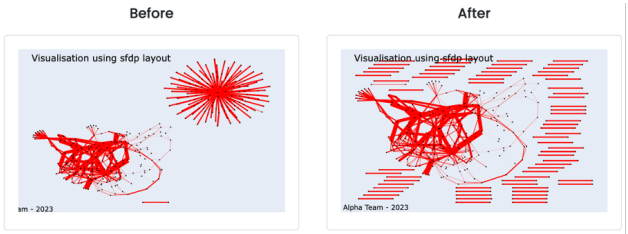

Figure 18: Lazarus Group's transaction network visualisation after removing the RAILGUN smart contract

Understanding the consequences of these custom attack scenarios can have significant implications for combating money laundering and illicit activities. A significant drop in global efficiency would make it more difficult for threat actors to obfuscate transactions and launder funds through centralised exchanges or other exit points. In contrast, the relative stability of local efficiency suggests that targeted disruption of key wallets and smart contracts may not be sufficient to completely dismantle their operations and may require more comprehensive measures.

The analysis of degree centrality reveals the presence of significant hubs within the network, which, when targeted, can disrupt the overall network structure. This can be valuable for identifying critical nodes within the network and devising targeted countermeasures to hinder threat actors' activities.

These insights can be applied to improve cybersecurity and blockchain analytics and develop countermeasures against money laundering and illicit activities. Possible applications include targeted sanctions and asset freezing, enhanced blockchain analytics, collaboration with centralised exchanges, regulatory improvements and enforcement, and public awareness and education. By leveraging the robustness analysis provided by our study, decision-makers

can make informed choices to enhance the security and integrity of the blockchain networks, ultimately benefiting society and the economy.

4.5 Deep Learning Embedding

4.5.1 Chinese Railway

We analyse the embeddings of the Chinese railway network using Node2Vec and Graph Neural Network (GNN) techniques to uncover the network's community and structure. Our goal is to understand the network's organisation, inter-community relationships, and individual node roles, as well as to explore real-life insights and applications based on these findings.

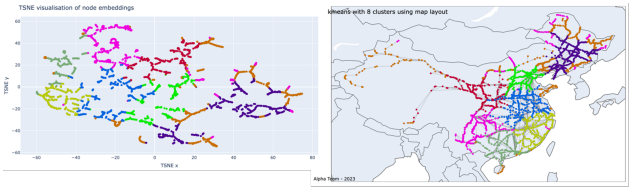

Figure 19: Chinese Railway network with Node2Vec embedding, P=1, Q=1

The Node2Vec technique demonstrates a strong community structure within the railway network, reflecting the regional nature of railway transportation. However, it does not reveal a clear structural embedding to differentiate between hubs, bridges, and peripheral nodes. On the other hand, the GNN approach (utilising the GraphSAGE model) provides a more meaningful structural embedding by incorporating input features such as degree centrality, triangle count, PageRank, and closeness centrality. This allows us to visually identify main hubs, bridges, traffic enhancers, and peripheral nodes in the network.

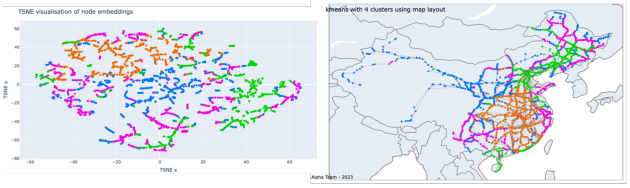

Figure 20: Chinese Railway network with Node2Vec embedding, P=0.25, Q=4

The insights derived from the analysis of the Chinese railway network using Node2Vec and Graph Neural Network (GNN) techniques can be applied to various real-life aspects of the railway system. By leveraging the knowledge gained from the robustness analysis and embeddings, decision-makers can make informed choices to enhance the performance and reliability of the railway network, ultimately benefiting society and the economy.

In the context of traffic congestion, the identification of critical nodes prone to congestion allows policymakers to prioritise infrastructure upgrades or develop alternative routes. This, in turn, can alleviate congestion and improve service reliability within the railway network.

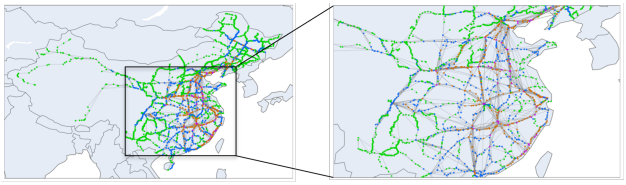

Figure 21: GNN structural embedding of the Chinese railway network

Regarding system resilience, understanding the community structure of the railway network can inform the development of contingency plans, resource allocation strategies, and emergency response coordination in the event of disruptions, such as natural disasters or accidents.

As for network expansion, the embeddings provide valuable information on potential opportunities for expanding the network and enhancing connectivity. By identifying underserved regions or nodes with high potential for growth, strategic investments can be made in infrastructure development, leading to a more balanced and accessible railway system.

In terms of maintenance, the insights about the roles of various nodes within the network can aid authorities in making informed decisions about resource allocation, such as where to station repair crews or maintenance equipment. Prioritising resources for critical nodes can ensure the efficient functioning of the network and minimise downtime caused by breakdowns or maintenance-related issues.

Lastly, analysing the community structure and node roles can help assess the environmental and socio-economic impact of the railway system. This information can be used to guide policies that promote sustainable development and social equity by evaluating how well the network serves various regions or population groups, identifying areas of potential improvement, and designing targeted interventions.

5 Future Works

In this paper, we have presented a fully operational and deployable software with a customised Docker image that handles dependencies and facilitates deployment. Despite its current capabilities, there are several potential improvements that could enhance the software's performance and functionality.

One such improvement is enhancing the backend through distributed API and parallelisation across different instances, ensuring a robust and efficient system, especially when multiple users perform clustering and deep learning analysis simultaneously. Additionally, the software could benefit from expanded capabilities by incorporating functionalities like link prediction or node prediction, enabling forecasting and providing further insights into complex network optimisation.

Another area of improvement involves integrating Cython to speed up the pre-processing of large-scale networks. This would enable our software to handle more complex networks efficiently without compromising usability or compatibility. Furthermore, developing functionality for cross-network analysis would offer users a comprehen-

sive understanding of complex systems involving multiple interconnected networks.

Lastly, designing a specialised pipeline customised for the efficient handling of massive networks could significantly improve the software's performance. This pipeline would focus on streamlined pre-processing, graph partitioning techniques, and the use of highly optimised algorithms and data structures tailored to large-scale graphs.

6 Conclusion

In conclusion, we have successfully developed and introduced an innovative software solution that democratises complex network analysis for a wide range of users. As complex networks continue to broadcast into various aspects of modern society, they have become indispensable in fields such as technology, transportation, and medicine. As a result, a user-friendly, flexible, and comprehensive software tool is essential for researchers, manufacturers, and commercial and financial professionals to leverage this knowledge effectively.

Our all-in-one application give opportunities to users to deeply examine the wide nature of complex systems. It features advanced visualisation, characterisation, and in-depth analysis of network structures and communities, leveraging machine learning and deep learning techniques to facilitate clustering, embedding, and visualisation of system behaviour.

Recognising the significance of resilience analysis in real-world applications, our software offers a comprehensive suite of analytical tools to assess the robustness of complex networks against various attack vectors. The insights and knowledge derived from our software have already proven invaluable insights in numerous applications.

For instance, our solution has aided blockchain analysts in unravelling the money laundering techniques employed by the Lazarus Group. By identifying critical wallets and smart contract protocols bridging illicit funds to fiat currency exit points, our software has enabled centralised exchanges and authorities to freeze assets and curb funding for these malicious entities.

Furthermore, our interactive resilience analysis has been instrumental in examining transportation networks, such as the Chinese Railway systems. By understanding the behaviour of this scale-free network and characterising key hubs and bridges, policymakers and managers can allocate resources more effectively, ensuring the continuity of high-quality service. With our solution, transportation network managers and stakeholders can facilitate network attacks simulation of the nodes and edges and see how the behaves proactively and prepare for uncertainties ahead.

Overall, our cutting-edge software solution significantly expands the accessibility and utility of complex network analysis, empowering stakeholders across various industries to harness the power of this valuable knowledge.

References

[Albert et al., 2000] Albert, R., Jeong, H., and Barabási, A. L. (2000). Error and attack tolerance of complex networks. *Nature*, 406:378–382.

[Apon and Saddik, 2019] Apon, A. and Saddik, A. E. (2019). Community detection in cryptocurrency networks using deep learning techniques. pages 81–86.

[Bai et al., 2020] Bai, Y., Huang, J., Wang, X., and Zhang, J. (2020). Machine learning for community detection in complex networks: A review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 12:e1521.

[Batty et al., 2019] Batty, M., Beecroft, M., Crooks, A., and See, L. (2019). Urban resilience: a review. *Geographical Journal*, 185:354–371.

[Bettencourt and Lobo, 2016] Bettencourt, L. M. A. and Lobo, J. (2016). The power of scaling in socio-economic systems. *Interface Focus*, 6:20150043.

[Boccaletti et al., 2006] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-Y. (2006). Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308.

[Cacciapuoti et al., 2020] Cacciapuoti, A. S., Bardi, A., and Penta, M. D. (2020). Community detection in the bitcoin network: An experimental analysis. *Journal of Network and Computer Applications*, 171:102740.

[Caldarelli et al., 2002] Caldarelli, G., Capocci, A., Rios, P. D. L., and Muñoz, M. A. (2002). Scale-free networks from varying vertex intrinsic fitness. *Physical Review Letters*, 89:258702.

[Chalkiadakis et al., 2022] Chalkiadakis, C., Perdikouris, A., and Vlahogianni, E. I. (2022). Urban road network resilience metrics and their relationship: Some experimental findings. *Case Studies on Transport Policy*, 10:2377–2392.

[Cutter et al., 2008] Cutter, S. L., Barnes, L., Berry, M., Burton, C., Evans, E., Tate, E., and Webb, J. (2008). The conceptual framework of vulnerability in social-ecological systems. *Environmental Hazards*, 7:15–27.

[Gao et al., 2019] Gao, H., Huang, Z., Zhou, J., Xu, Y., and Cui, Z. (2019). Explainable graph neural networks via attention-based graph explanation. *arXiv preprint arXiv:1905.13686*.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826.

[Gonzalez and Hong, 2019] Gonzalez, M. C. and Hong, Y. (2019). Resilience metrics for transportation systems: a critical review. *Transport Reviews*, 39:233–252.

[Hamilton et al., 2017] Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs.

[He et al., 2020] He, L., Xu, Y., Li, H., and Li, Y. (2020). Unsupervised community detection in bitcoin transaction networks via convolutional autoencoder. pages 1–6.

[Hu and Wu, 2019] Hu, Z. and Wu, Y. (2019). Airline network community detection based on k-means and hierarchical clustering. *Journal of Advanced Transportation*.

[Ip and Wang, 2011] Ip, W. H. and Wang, D. (2011). Resilience and friability of transportation networks: Evaluation, analysis and optimisation. *IEEE Systems Journal*, 5:189–198.

[Jiang et al., 2021] Jiang, Y., Xie, X., and Feng, Y. (2021). Predicting connectivity and passenger flows in airline networks with machine learning. *Transportation Research Part C: Emerging Technologies*, 124:103196.

[Jin et al., 2019] Jin, D., Xie, Y., Zhang, X., and Wang, Y. (2019). Hierarchical clustering-based community detection in cryptocurrency networks. *Journal of Parallel and Distributed Computing*, 129:70–79.

[Jin et al., 2018] Jin, P., Zhang, Y., Wang, Y., and Zhang, F. (2018). Network resilience analysis: a literature review with a focus on transportation networks. *Transport Reviews*, 38:52–73.

[Jing et al., 2019] Jing, Y., Wu, Y., and Yan, W. (2019). Community detection in urban road network using random forest. *ISPRS International Journal of Geo-Information*, 8:338.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks.

[Kumar and Garg, 2019] Kumar, S. and Garg, S. (2019). Spectral clustering based community detection in bitcoin transaction networks. *PloS one*, 14:e0210117.

[Li et al., 2021] Li, J., Wang, Y., and Zhang, Y. (2021). A deep learning approach for community detection in bitcoin transaction networks. *IEEE Access*, 9:54267–54276.

[Li et al., 2018] Li, Y., Yu, R., Shahabi, C., Liu, Y., and Yang, J. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

[Liao et al., 2020] Liao, M., Yang, X., and Zhang, X. (2020). Community detection in bitcoin transaction networks based on a hybrid method. *Complexity*, 2020.

[Moutsinas et al., 2021] Moutsinas, G., Zou, M., and Guo, W. (2021). Uncertainty of resilience in complex networks with nonlinear dynamics. *IEEE Systems Journal*, 15:4687–4695.

[Naimi et al., 2019] Naimi, B., Rodriguez-Galiano, V., and Campos-Celador, A. (2019). Automatic community detection in urban road networks with support vector machines. *Computers, Environment and Urban Systems*, 74:38–47.

[Newman, 2006] Newman, M. E. (2006). The structure and function of complex networks. *https://doi.org/10.1137/S003614450342480*, 45:167–256.

[Rasouli et al., 2018] Rasouli, S., Ramezani, M., Liu, J., and Wang, Z. (2018). Road network resilience analysis using accessibility-based cumulative cutting methodology. *Transportation Research Part A: Policy and Practice*, 118:198–211.

[Ravasz et al., 2002] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555.

[Shen et al., 2018] Shen, X., Zhu, Y., Li, L., and Liu, P. (2018). A dl-based approach for anomaly detection in bitcoin transaction networks. *IEEE Access*, 6:71041–71048.

[Smith et al., 2011] Smith, P., Hutchison, D., Sterbenz, J. P., Schöller, M., Fessi, A., Karaliopoulos, M., Lac, C., and Plattner, B. (2011). Network resilience: a systematic approach. *IEEE Communications Magazine*, 49:88–97.

[Sun et al., 2020] Sun, W., Bocchini, P., and Davison, B. D. (2020). Resilience metrics and measurement methods for transportation infrastructure: the state of the art. *Sustainable and Resilient Infrastructure*, 5:168–199.

[Sun et al., 2021] Sun, X., Li, J., Li, W., Li, C., Yang, H., and Liu, Y. (2021). Community detection in bitcoin transaction networks using a multi-objective optimization algorithm. *IEEE Access*, 9:40503–40514.

[Veličkovi'c et al., 2018] Veličkovi'c, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph attention networks.

[Vigna and Francesconi, 2019] Vigna, G. and Francesconi, E. (2019). Community detection in blockchain-based networks: a case study in ethereum. pages 1–6.

[Wang et al., 2020] Wang, Y., Liu, H., Sun, Y., and Chen, G. (2020). A review of graph neural networks for network analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10:e1407.

[Wang et al., 2021] Wang, Y., Ma, Z., Liu, H., Zhang, W., and Li, X. (2021). Community detection in cryptocurrency networks based on multilayer networks. *Plos one*, 16:e0246364.

[Wang et al., 2019] Wang, Z., Zhan, X., Zhang, Y., and Cheng, Y. (2019). Railway station community detection based on support vector machine. *IEEE Access*, 7:52678–52686.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.

[Wu et al., 2021] Wu, J., Li, Y., Sun, Y., and Li, J. (2021). Graph-based community detection in ethereum transaction networks. *Journal of Parallel and Distributed Computing*, 154:211–221.

[Wu et al., 2020] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24.

[Xie et al., 2020] Xie, J., Kelley, S., and Szymanski, B. K. (2020). A comparison of community detection algorithms on artificial networks. *Scientific Reports*, 10:1–19.

[Yan et al., 2018] Yan, J., Cui, H., Guo, W., Zhu, X., Wang, W., Yu, L., Lu, J., and Sun, H. (2018). Quantitative analysis of the resilience of complex systems with

nonlinear dynamics under uncertainty. *Reliability Engineering System Safety*, 178:20–34.

[Yang et al., 2020] Yang, Y., Zhang, Y., Han, X., and Chen, B. (2020). Community detection in road networks based on graph convolutional networks. *IEEE Access*, 8:129658–129666.

[Zeng et al., 2021] Zeng, J., Li, Z., Li, X., and Luo, L. (2021). Community detection in railway network based on graph convolutional networks. *Neural Computing and Applications*, 33:6987–6998.

[Zeng et al., 2020] Zeng, Y., Chen, J., and Li, S. (2020). Community detection in blockchain networks: A survey. *Journal of Network and Computer Applications*, 167:102786.

[Zhang et al., 2019] Zhang, R., Wang, W., and Yang, C. (2019). Community detection in bitcoin transaction networks based on a fast louvain algorithm. *IEEE Access*, 7:93057–93066.

[Zhang et al., 2021] Zhang, Y., Yang, Y., and Chen, B. (2021). Community detection in transportation networks based on graph convolutional networks. *IEEE Transactions on Intelligent Transportation Systems*, 22:4343–4353.

[Zhou et al., 2020] Zhou, L., Guan, X., Yu, W., and Wang, S. (2020). Community detection in railway networks based on convolutional neural network. *Neural Computing and Applications*, 32:8469–8479.