# Fraud Detection on a Healthcare Network, Using Complex Network Analysis

Santhosh Kumar Rajamani[1], and Radha Srinivasan Iyer[2]

[1] MAEER MIT Pune's MIMER Medical College and DR. BSTR Hospital, Talegaon D, Pune, Maharashtra, India https://orcid.org/0000-0001-6552-5578
minerva.santh@gmail.com

[2] Neuro-physiotherapist, SEC Centre for Independent Living, Naigaon, Pune, MH, India https://orcid.org/0000-0001-7387-4401
Rads.physio@gmail.com

**Abstract.** This compilation describes the various Complex network analytical techniques to detect fraud on a healthcare network. These tactics can be used by admins for surveillance and detection of crime. A motif is a small, organized structure within a network that repeatedly, appears in a network. The second part focusses on using various Network Motifs like feedback loop, clique motifs, rogue nodes and to detect fraudulent activities. These strategies are illustrated using a few simplified fraud-detecting Python programs using NetworkX.

**Keywords:** networks in healthcare, network analysis, network motifs, fraud detection, machine learning, complex network analysis, python, networkx.

## 1 Introduction

Fraud detection using network analysis is an important tool for detecting potential financial fraud cases. The method considers the links between the entities in a data set and uses pattern recognition techniques to identify any irregularities that may point to fraudulent activity. By using both qualitative and quantitative measures, fraud supervisors can identify which accounts are connected to each other and potentially linked with fraudulent behavior patterns. Moreover, this technique helps uncover suspicious relationships between money transfers, client profiles, service providers, locations and payment methods that may indicate a fraudulent case. Overall, this method of fraud detection aids investigators in identifying new suspects as well as protecting companies from existing threats in a much more efficient manner compared to traditional methods [1].

## 2 Complex Network Analytics to detect fraud

To find fraud on a network, several techniques can be applied. Many typical tactics include:

**Anomaly detection**. Finding unexpected rhythms of activity or transactions that can point to deception is known as anomaly detection [2].

**Link analysis.** Link analysis entails examining the relationships between people or things in a network to find trends that might be a sign of fraud. Link analysis is a method for examining the connections among the items in a network. By examining patterns of linkages between network entities and recognising unusual or suspect patterns, it can be used to detect fraudulent behaviour on a network. Either physical labour or specialist software tools can be used to do this. This may entail the need to collect and analyse data on the links between entities in the network, such as communication records, financial transactions, and other sorts of data, to do a link analysis for the aim of identifying fraud on a network. Afterward, this information can be used to produce graphic depictions of the connections between items, like graphs or network diagrams. Suspicious links are those that exhibit, unusually high or low engagement levels between specific nodes, or unexpected or strange relationships between nodes, or resource flows between communities that are out of proportion and (deceitful) nodal behaviour patterns or transactions that recur over time [3,4].

***NetworkX for link analysis***. The structure and characteristics of the network can then be examined using a range of NetworkX functions and techniques. The following are a few examples of connection analysis tasks that NetworkX can carry out: examining the network's connectedness, such as by finding the shortest route between nodes or figuring out which nodes are most connected to other nodes finding anomalies or odd patterns in the network, for example, by contrasting the observed network attributes with those of a random network With NetworkX, you can carry out a wide range of additional link analysis tasks. The techniques and algorithms employed will depend on the demands of the analysis [5, 6].

**Social network analysis**. Social network analysis entails looking at the connections between people or things in a social network to spot any trends that might point to fraudulent behaviour [7].

**Machine learning algorithms to detect fraud.** Machine learning is the process of applying algorithms to examine network data and spot trends that might be signs of fraud. For spotting network fraud, machine learning can be a useful technique. Machine learning algorithms can learn to recognise patterns that can be predictive of future fraudulent behaviour by studying data from previous fraudulent behaviour. Then, for further examination, these algorithms can be utilised to highlight possibly fraudulent transactions or behaviour. For detecting fraud on a network, numerous alternative machine learning techniques can be applied. Typical examples include:

1.*Decision trees algorithms* build models using a succession of decisions, with each branch of the tree denoting a potential result.

2. *Neural network algorithms* that draw their design cues from the structure and operation of the human brain. They excel at finding patterns in data.

3. *Support Vector Machines (SVMs*) which are frequently used for classification problems, define a border between several classes of data.

It is crucial to remember that no single machine learning method is suitable for uncovering every kind of fraud on a network. Instead, a sagacious combination of algorithms might be required to accurately detect and stop fraud [8].

## 4 Measures of Centrality in Network analytics

The various measure of centrality of a vertex or node provides information on the role and relative weight of a node relative other nodes in a complex network. The following are measures of centrality.

**Betweenness Centrality**. A graph's vertex's centrality is determined by its *betweenness centrality*. It is determined by counting how many shortest routes exist between each vertex and every other vertex. Because it resides on numerous shortest paths, a vertex with high betweenness centrality has a significant impact on the flow of information or resources through the graph. Think of a graph of cities connected by roads as an illustration. Since many road journeys pass through a city that is situated on a key transportation route between other cities, its *betweenness centrality* may be high. Both directed and undirected graphs can calculate *betweenness centrality* [9].

**Node strength centrality**. *Node strength centrality*, which is based on the quantity and weight of a node's connections to other nodes, is a measure of a node's significance in a network used in complex network analysis. It is described as the total of all edge weights occurring to the node. It is possible to determine the node strength centrality for both directed and undirected graphs. With many connections to other nodes and/or connections with high weight, a node with a high strength centrality score has many connections overall. This trait can be helpful in several applications, such as locating crucial infrastructure in a transportation network or recognising essential nodes in a social network [9].

**Clustering centrality**. *Clustering centrality* is a measure of a node's centrality in a graph used in Complex network analysis that is based on how closely related those nodes are to one another. It is described as the percentage of neighbouring pair connections between the nodes. It is possible to determine clustering centrality for both directed and undirected graphs. With numerous neighbours who are connected to one another and a high clustering centrality score, a node is well-connected to the other nodes in the graph. Applications for this trait include locating communities inside a network or recognising important nodes in social networks [10].

**Proximity Centrality of a Node**. According to Complex network analysis, the *proximity centrality of a node* in a graph is determined by the average length of the shortest paths connecting it to every other node in the graph. It is described as equal to the reciprocal of the total of the distances between each node in the graph. Graphs that are directed or undirected can both have their closeness centrality computed. A node that has a high proximity centrality score is well-connected to other nodes and has many short pathways to them. This trait can be helpful in several applications, such as locating crucial infrastructure in a transportation network or recognising essential nodes in a social network [10].

## 3 Network Motifs in Networks to detect fraudulent activities

A motif is a small, organized structure within a network that repeatedly, appears in a network. A motif in graph analysis is a brief subgraph that reappears regularly in a larger graph. Motifs can be used to locate structures and patterns within a graph and to shed light on the connections and interactions between its constituent parts. An illustration of a prevalent pattern in a social network graph would be a trio of individuals linked together by friendships. Finding and examining motifs can assist scholars in comprehending a graph's fundamental structure and operation. Dyads (2 vertices) and triads (3 vertices) are examples of small subgraphs or motifs that are made up of a collection of nodes and links between them [11].

Network motifs are connectivity patterns that show up in a network more frequently than would be predicted by chance. In the investigation of complex systems, such as social and biological networks, they are frequently employed. Network motifs may be utilised in crime analysis to spot linkages or patterns of behaviour among people or groups that might be involved in criminal activities. There is a chance that some network motifs are more common in fraudulent networks than in genuine ones. For instance, in a fraud network where the central "hub" character is orchestrating the fraudulent activities while others are taking part, a motif with several people who are all related to one another may be more likely to arise [11].

Building a network that represents the connections or interactions between people or groups suspected of engaging in criminal conduct is one method of employing network motifs for crime analysis. The network might be built using data from phone records, social media interactions, or other kinds of association or communication data. Comparing the motifs found in the network to those found in other networks or to recognised patterns of criminal activity is another method for evaluating network motifs for crime analysis. For instance, it may indicate that a motif is significant in the context of drug trafficking networks if a network of people suspected of being involved in drug trafficking displays a specific sort of theme that has been seen in other networks of drug traffickers. Using algorithms, it is possible to find network motifs in the data after the network has been built [12].

Comparing the network motifs to a null model, a fictitious network produced by a random process that retains some aspects of the actual network (such the number of nodes and edges) but eliminates any innate structure or patterns, is one technique to study the network motifs. It may indicate that patterns are significant in the context of the network under study if they appear more frequently in the real-world network [11].

For instance, it may indicate that a motif is significant in the context of drug trafficking networks if a network of people suspected of being involved in drug trafficking displays a specific sort of theme that has been seen in other networks of drug traffickers. It is important to keep in mind that network motifs are just one tool that may be utilised in the study of complex systems; to provide a more thorough knowledge of the system being investigated, they should be used in conjunction with other techniques and information sources [12].

By locating patterns of behaviour or connections between people or entities engaged in fraudulent activities, network motifs can be utilised to detect fraud. Based on information from records like bank statements, credit card statements, or other financial records, for instance, a network reflecting financial transactions might be built. To find anomalies or patterns that might point to fraudulent activity, algorithms could be used to identify network motifs within the data [12]. These motifs could then be compared to known patterns of fraudulent behaviour.

**Clique motif.** A group of nodes in a network that are all linked to one another is referred to as a *clique*. This is a collection of interconnected nodes, is another form of network motif that could be helpful in the identification of fraud. This trend can point to a network of people or organisations that are conspiring to perpetrate fraud.

An example of a clique motif in network analysis is a collection of three or more connected nodes that together make up a whole subgraph. Because they can provide light on a network's structural characteristics and the ways in which various groups of nodes interact with one another, clique motifs are frequently explored in network research. For instance, the presence of clique patterns in a network may be a sign of close-knit communities or the emergence of alliances between various node groupings. Clique motifs can occasionally be used to pinpoint certain pivotal roles in a network including fraud perpetuation.

For instance, nodes that belong to several cliques may be thought of as playing a "hub" role in the network, bridging several communities. Similarly, this, nodes with limited connections to other nodes in the network are said to be "peripheral" if they only belong to one clique [11].

**Rogue node motif.** A vertex or device connected to a network but not connected to any other nodes is referred to as a rogue node. A rogue node is of zero degree as it has no edges connecting it to other parts of the network. These nodes could end up being exploited to carry out harmful tasks like hacking, virus distribution, or other types of criminality, which could endanger the network's security. Using network modelling and simulation tools is one way to examine how rogue nodes affect a network. With the help of these tools, admins can develop virtual representations of the network and test various scenarios to determine how they can impact its performance and security. Users may be more susceptible to attack from rogue nodes, for instance, if they click on links in questionable emails and do not manage their passwords according to recommended standards [12].

**Feedback loop motif.** A feedback loop is a loop or circular edge that begins and finishes at the same node in a graph. The "feedback loop" is a pattern in which a node in the network has both inbounded and outgoing connections to the same other node, is one form of network motif that might be particularly pertinent in the context of fraud detection [13].

Feedback loops can be used to detect network anomalies by observing the behavior of the network over time. Feedback loops measure the input and output of the network and compare them to a predefined baseline. If the output deviates from the baseline, it could indicate an anomaly. Additionally, feedback loops can be used to detect anomalies in traffic patterns, as they can observe the time between successive packets and measure the rate of change in the data. By monitoring the data over time, feedback loops can identify any sudden or unexpected changes that could indicate an anomaly [14].

This pattern might point to a plan in which one party assumes several identities to scam others. Feedback loops are common in many biological networks and play vital role in maintenance of Network state, a typical example is 'circadian rhythm,' or 24-hour periodic sleep wake cycle of Drosophila which is driven by a positive feedback loop between accumulation and degradation of 2 proteins namely: Per, and Tim. These loops are concerned with all or none decision making in biological networks, for example in the above case a fruit-fly cell cannot be awake and sleep at the same time [14].

Feedback loops in Complex network analysis are used to examine the interactions between two or more components of a system. A feedback loop is a process in which the output of a system is fed back as input into the same system. In complex analysis, feedback loops can help identify emerging patterns and trends, as well as uncover hidden relationships between variables. Feedback loops can also be used to identify potential problems with a system, such as a lack of efficiency or an unexpected increase or decrease in output [15].

## 2 NetworkX Python Module for Complex Network Analysis

NetworkX is a Python library for analysing complex networks and graph algorithms. It offers numerous tools and techniques for network analysis and can be applied to link analysis. It can be used to detect fraud by analysing the network structure of a system. NetworkX can be used to identify clusters of nodes that are connected to each other or to other nodes in the network. It can also detect anomalies by comparing the structure of the current network to that of a baseline network. Additionally, NetworkX can be used to detect patterns of fraudulent activity by analysing the connections between nodes and the flow of data within the network. NetworkX is optional module available via PyPi channel, which provides Complex network analysis capabilities in Python 3.5 and above. This module must be installed via python package manager pip (python –m pip – install networkx or just pip –install networkx). The "simple_cycle" is a NetworkX method can also be used to locate every feedback loop in a graph. A self-loop, or cycle that connects a node to itself, can be detected using NetworkX. This method returns a generator that produces lists of nodes representing the feedback loops [16]. Here is a simple illustration of how you can apply this technique in Complex network analysis using NetworkX

```
# Modules networkx, matplotlib, numpy and PySimpleGUI have to be installed
via pip pip -install networkx
import networkx as nx
import matplotlib.pyplot as plt
import PySimpleGUI as sg
sg.theme("material 2")
# Create a graph with multiple feedback loops
G = nx.DiGraph()
G.add_edge('Vijay', 'Suresh')
G.add_edge('Suresh', 'Nirav')
G.add_edge('Nirav', 'Vijay')
# Find all the feedback loops in the graph and show them to user via Popup
box
for loop in nx.simple_cycles(G):
   sg.popup(loop, title="feedback loops in the graph by simple_cycles",
auto_close=15)
# Create figure and plot for preparation
fig = plt.figure(figsize=(20,10))
# Create the Options dictionary for plotting
options = {'node_color': 'yellow','node_size': 700,'alpha':0.9,'width': 1,
    'edge_color':'red',}
# Plot the network using Kamada Kawai layout algorithm
nx.draw_kamada_kawai(G, with_labels=True,**options)
# Supply a clear title and show the network
plt.title("Feedback Loops in Criminal network of 3 persons Vijay, Suresh,
and Nirav")
plt.show()
```

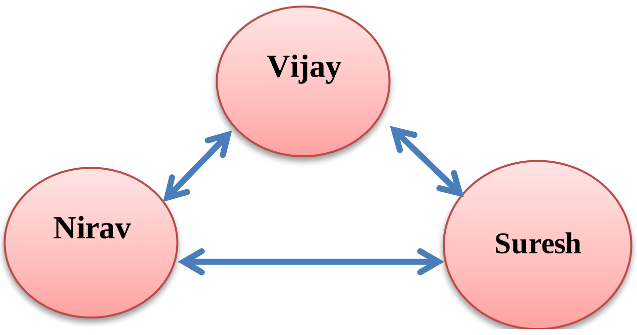

**Fig. 1.** Feedback Loops in Criminal network of 3 persons Vijay, Suresh, and Nirav. The feedback loops are self-propagating, like so Nirav bribes the bank official who in turn bribes the auditor Vijay who gives a “cut” of his income back to “Nirav”. Thus, the criminal network grows in strength by feedback loops [17].

Another loop method called as *nx.simple_cycles* module can be used to determine whether a graph contains a feedback loop [17]. If at least one feedback loop is present in the graph, this method returns True; otherwise, it returns False.

```
# Modules networkx, matplotlib, numpy and PySimpleGUI have to be
installed via pip pip –install networkx
import networkx as nx
import matplotlib.pyplot as plt
import PySimpleGUI as sg
sg.theme("material 2")
edges = [('Home', 'Home'), ('Home', 'Office'),('Office','Home'), (
'Club','Home'), ('Home','Club'), ('Club', 'Home'), ('Club', 'Office'),
('Club', 'Club'),('Farm','Home'),('School','Home'),('Farm','Farm')]
#Create a directed graph
G = nx.DiGraph(edges)
#Detects loops in Original Orientation they appear in the graph
sg.popup(nx.find_cycle(G, orientation="original"))
#Detects loops in but ignore the Orientation they appear in the graph
sg.popup(list(nx.find_cycle(G, orientation="ignore")))
# Create figure and plot for preparation
fig = plt.figure(figsize=(20,10))
# Create the Options dictionary for plotting
options = {'node_color': 'yellow','node_size':
700,'alpha':0.9,'width': 1,
    'edge_color':'red',}
# Create a shell layout visualization
nx.draw_shell(G, with_labels=True,**options)
plt.show()
```

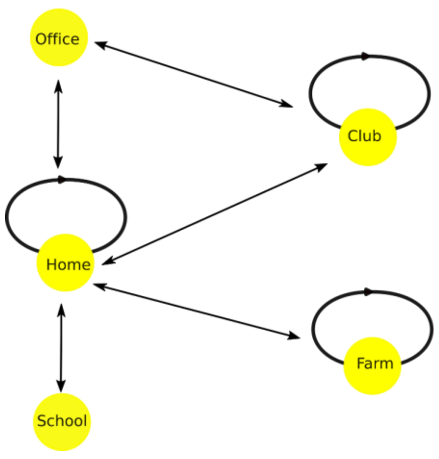

**Fig. 2**. Another example of Feedback Loops detection in Python NetworkX. This time Orientation of the loops is also considered and ignored by command nx.find_cycle (G, orientation="original")). It can be observed that Home, Club, and farm have self-loops, in addition Home, Club and Office nodes form a loop. NetworkX can greatly simplify a variety of Complex network analytical tasks

## 3 Conclusion

As a summary, the use of network science and machine learning can be a powerful tool in the fight against healthcare fraud, helping to identify and prevent fraudulent activity and ensure that healthcare resources are used efficiently and effectively.

## References

1. Coderre, D.: Fraud Detection Using Digital Analysis. EDPACS. 27, 1–8 (1999). https://doi.org/10.1201/1079/43249.27.3.19990901/30268.1.
2. Vaddella, V.R.P., Rachakulla, S.: Two-Stage Opportunistic Sampling for Network Anomaly Detection. International Journal of Computer and Electrical Engineering. 1068–1076 (2010). https://doi.org/10.7763/ijcee.2010.v2.277.
3.Wang, H., Li, Y., Guo, K.: Countering Web Spam of Link-based Ranking Based on Link Analysis. Procedia Engineering. 23, 310–315 (2011). https://doi.org/10.1016/j.proeng.2011.11.2507.
4.Vaddella, V.R.P., Rachakulla, S.: Two-Stage Opportunistic Sampling for Network Anomaly Detection. International Journal of Computer and Electrical Engineering. 1068–1076 (2010). https://doi.org/10.7763/ijcee.2010.v2.277.
5. Dmitry Zinoviev, Adaobi Obi Tulton: Complex network analysis in Python: recognize - construct - visualize - analyze - interpret. The Pragmatic Bookshelf, Raleigh, North Carolina (2018).
6. Kollu, V.V.R., Amiripalli, S.S., Jitendra, M.S.N.V., Kumar, T.R.: A Network Science-based performance improvement model for the airline industry using NetworkX. International Journal of Sensors, Wireless Communications and Control. 10, (2020). https://doi.org/10.2174/2210327910999201029194155.
7. Alhajj, R., Memon, N.: Introduction to the second issue of Social Network Analysis and Mining journal: scientific computing for social network analysis and dynamicity. Social Network Analysis and Mining. 1, 73–74 (2011). https://doi.org/10.1007/s13278-011-0022-z.
8. Khorram, T.: Network Intrusion Detection using Optimized Machine Learning Algorithms. European Journal of Science and Technology. (2021). https://doi.org/10.31590/ejosat.849723.
9. Bornholdt, S., Heinz Georg Schuster: Handbook of Graphs and Networks. John Wiley & Sons (2006).
10. Mei, G., Tu, J., Xiao, L., Piccialli, F.: An efficient graph clustering algorithm by exploiting k-core decomposition and motifs. Computers & Electrical Engineering. 96, 11.7564 (2021). https://doi.org/10.1016/j.compeleceng.2021.107564.
12. Ismail, S., El Mrabet, Z., Reza, H.: An Ensemble-Based Machine Learning Approach for Cyber-Attacks Detection in Wireless Sensor Networks. Applied Sciences. 13, 30

(2022). https://doi.org/10.3390/app13010030.

13. Isogai, T.: Dynamic correlation network analysis of financial asset returns with network clustering. Applied Network Science. 2, (2017). https://doi.org/10.1007/s41109-017-0031-6.

13. Gould, R.: Graph theory. Dover Publications, Inc, Mineola, New York (2012).

14. Estrada, E.: The structure of complex networks: theory and applications. Oxford University Press, New York (2012).

15. Dehmer, M.: Structural Analysis of Complex Networks. Springer Science & Business Media (2010).

16. Platt, E.L.: Network Science with Python and NetworkX Quick Start Guide. Packt Publishing Ltd (2019).

17. Rogel-Salazar, J.: Advanced Data Science and Analytics with Python. CRC Press (2020).

18. Chartrand, G., Zhang, P.: A First Course in Graph Theory. Courier Corporation (2013).