Journal of Discrete Mathematics and Its Applications 11 (2) (2026) 131–151

Journal of Discrete Mathematics and Its Applications

Available Online at: http://jdma.sru.ac.ir

Shahid Rajaee Teacher Training University

Research Paper

Graph-enhanced fraud detection in health insurance: integrating centrality metrics and community analysis for improved accuracy

Mohammad Mahdi Ahmadi[1], **Asma Hamzeh**[2,*], **Ali Kamandi**[1]

[1] Department of Engineering Sciences, University of Tehran, Tehran, I. R. Iran

[2] Department of Modern Insurance Technologies, Insurance Research Center, Tehran, I. R. Iran

Academic Editor: Bo Zhou

Abstract. The dynamic nature of fraud and the emergence of new fraudulent methods in the insurance industry, along with the insufficient capability of insurance companies to prevent, identify, and combat it, may lead insurance companies toward bankruptcy in the not-too-distant future. Given that insurance companies serve as the main entity providing insurance services and compensating losses, their significant role in preventing and detecting fraud makes the various strategies they employ to counter fraud significant. In the current situation, identifying and analyzing the phenomenon of fraud, which leads to increased costs in the insurance industry, appears to be essential for controlling factors and ensuring the survival of insurance companies.

This article examines the use of graph theory for fraud detection in the health insurance industry. By extracting information from relevant databases and constructing a comprehensive network graph, existing patterns are analyzed and suspicious fraudulent cases are identified. The proposed method was implemented on real data, and the results showed that this method is capable of detecting fraud with 95% accuracy, which is an improvement over the existing method. All implementation codes and supplementary materials are openly available on GitHub (`https://github.com/MohammadMehdi41/fraudDetection.git`).

Keywords. graph theory, health insurance, fraud detection.
Mathematics Subject Classification (2020): 05C82, 68T05, 68T09, 91G70.

*Corresponding author (Email address: hamzeh@irc.ac.ir)
Received 11 December 2025; Revised 07 January 2026; Accepted 16 March 2026
First Publish Date: 01 June 2026

1 Introduction

Health insurance is one of the most important and essential types of insurance available and one of the critical services needed by individuals in a community. However, one of the problems currently plaguing this industry is organized fraud by medical teams, patients, and hospitals. A vast amount of fraud annually threatens this industry. For instance, trillions of dollars of fraud occur in the United States every year. In these frauds, healthcare, medical, and pharmaceutical service centers such as hospitals, clinics, laboratories, physicians, and pharmacies typically participate, along with insured patients on the other side. Insurers usually have to examine a considerable amount of medical documentation to detect this fraud, which is time-consuming, and given that the frauds occur in an organized manner, uncovering and reviewing this documentation becomes doubly challenging [7].

Fraud is any intentional act or omission that is designed to deceive and mislead others, causing harm to its victims while enabling the perpetrators to gain benefits [15]. Fraud leads to increasing problems such as losses for insurance companies. Since insurance is one of the key organizations in the economic status of a country, it is necessary to consider actions to identify fraud [12].

Fraudulent behavior by healthcare service providers, patients, and insurance company staff, imposing unnecessary costs, has become a serious problem for insurance systems. Insurance organizations, after receiving expense documents from healthcare providers, including treatment centers, physicians, pharmacies, and laboratories, begin the process of reviewing expenses and reconciling them with established criteria and cost tables, a process known as document handling. In manual handling by experts within organizations, limitations such as human error, lack of expert human resources, time constraints for human activities, inconsistency in handling quality, potential interactions between evaluators and evaluates, and other factors impact the handling process [1].

To achieve more effective fraud detection methods, many researchers have recently proposed more complex anti-fraud methods based on data mining, machine learning, and other analytical techniques. These new methods have major advantages, such as the automatic learning of fraud patterns from data, probability detection of fraud for each case, and identification of new types of fraud, leading to the creation of an effective method with higher accuracy, reduced costs, and decreased computational time [5]. Accordingly, this article presents an effective and optimized solution for fraud detection in insurance using a combination of data mining algorithms and graph theory. This paper is organized in a way that initially explains the issue of fraud, including examples of fraud, followed by a review of the research background on fraud detection. It then states the mathematical preliminaries of the model and proposed algorithm, describes the implementation of the algorithm, and finally presents the conclusion.

This study proposes an empirical, end-to-end evaluation framework that integrates graph construction from insurance claim interactions with machine-learning models. The contribution of this work lies in demonstrating how graph-based structural features and community detection can enhance fraud detection performance through systematic comparative experi-

ments.

2 Theoretical Foundations

A network, also known as a graph in mathematics, consists of a non-empty set of objects called vertices (represented by V) and a set of edges that connect the vertices, represented by E. Such a graph is denoted by $G = (V, E)$ and is referred to as a simple graph. In a simple graph, there is at most one edge between any two vertices.

Graph analysis, also called network analysis, is the study of relationships between entities such as customers, products, devices, and so on. Additionally, organizations use graph models to gain insights that can be utilized in marketing or social networks. Graph analysis has gained significant importance due to the growing market demand. In this paper, the graph diagram includes vertices representing physicians and edges representing claims. There is a wide range of metrics that can be calculated for each vertex representing physicians. Depending on the purpose of this research, the focus will be on vertex degree, closeness centrality, eigenvector centrality, and PageRank, which primarily indicate the centrality and influence of a vertex in the graph. Their definitions are elaborated below.

2.1 Degree

Degree indicates the number of edges that connect to a vertex.

2.2 Closeness Centrality

As the name suggests, closeness centrality measures the proximity and centrality of a vertex to other vertices. For a given vertex u, it represents the reciprocal of the sum of the shortest path distances from u to all other $n-1$ vertices:

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(u,v)},$$

where $d(u,v)$ is the shortest distance between u and v, and n is the number of vertices in the graph.

2.3 Eigenvector Centrality

Eigenvector centrality takes the centrality of a vertex based on the centrality of its neighbors. This means that a vertex with a high score is also connected to other influential vertices with high scores.

For a given graph, the adjacency matrix A is defined such that $a_{uv} = 1$ if vertex u is connected to vertex v, and otherwise the matrix entry is zero. The relative centrality x for vertex v can be defined as follows:

$$x_v = \frac{1}{\lambda} \sum_{t \in G} a_{(v,t)} x_t.$$

It can also be rewritten as the eigenvalue equation:

$$Ax = \lambda x.$$

The component v of x provides the relative centrality score of node v in the network. More information on this score and its calculation can be found in [2].

2.4 PageRank

This score relies on the normalized eigenvector centrality or normalized prestige p, which is calculated using the normalized adjacency matrix N defined as follows:

$$N(u,v) = \begin{cases} \frac{1}{od(u)} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases},$$

where $od(u)$ is the out-degree of vertex u.

The normalized prestige score $p(v)$ of a vertex v is calculated by solving the following equation:

$$p(v) = \sum_u N^T(u,v) \cdot p(u).$$

More information can be found in [14].

The use of graph-derived features in fraud detection is theoretically motivated by known behavioral patterns in healthcare networks. Fraudulent actors often occupy structurally abnormal positions, either excessively connected to unrelated entities or isolated within suspicious subgroups. Degree centrality reflects interaction volume, eigenvector centrality captures influence through highly connected neighbors, and PageRank identifies nodes receiving disproportionate attention within the network. These graph characteristics are commonly associated with irregular provider-patient relationships and justify their inclusion in the proposed framework.

2.5 Louvain Method

The Louvain method is an optimization algorithm for community detection in large networks. The primary goal of this algorithm is to partition the network into communities or clusters in which internal connections are denser and stronger than external ones. This is achieved by optimizing a measure called the modularity score.

Community detection using the Louvain algorithm provides an interpretable mechanism for identifying groups of physicians and agencies that exhibit similar behaviors. In many fraud scenarios, small clusters of providers tend to cooperate or repeatedly share patients in unusual patterns. Therefore, the community labels generated in the third method act as structural indicators of potential collusion. This conceptual justification strengthens the model design without altering the previously reported accuracy values.

The mathematical definition of the modularity score is as follows:

$$Q = \frac{1}{2m} \sum_{vw \in E} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_i, c_j),$$

where the Kronecker delta function is equal to 1 if i and j belong to the same community and 0 otherwise. In addition, m represents the number of edges, A_{vw} is the adjacency matrix, k_v is the degree of vertex v, and E is the set of graph edges.

The Random Forest algorithm is a supervised machine learning algorithm widely used for classification and regression problems. Increasing the number of trees generally improves the predictive capability and robustness of the model. Random Forest consists of several decision trees built on different subsets of the dataset, and the final prediction is obtained by averaging or voting across the trees. The algorithm performs effectively on large datasets and can produce accurate predictions even in the presence of incomplete information [10].

Graph-based structural metrics can reveal coordinated or unusual behavioral patterns that are not detectable using tabular features alone. High-degree physicians or agencies may indicate unusually dense interactions, while high PageRank or eigenvector centrality values may reflect influential or suspiciously interconnected actors. Such graph-based patterns are often associated with organized fraud rings and therefore provide strong theoretical justification for incorporating graph features into the proposed fraud detection framework.

3 Research Background

Community detection plays an important role in graph theory and has numerous applications. Community detection is widely used in machine learning applications. Among networks, connections can take various forms such as friendship, consultation, shared interests, or even trust. Paper [8] provides a comprehensive overview of community detection algorithms. One of the most popular algorithms uses modularity as an objective function to optimize cluster assignments until an optimal structure is reached [13]. These algorithms are effective for decomposing the entire network of physicians into smaller communities based on their similarities.

The idea of using graphs for fraud detection in the health insurance domain is discussed in papers such as [2,4,11]. These papers address the significance of graph analysis and the value of graphs for fraud detection, with the way in which graph analysis is conducted differing from one research work to another, and each approaching this task differently. In addition, there have been many previous studies on fraud detection using graph analysis in other domains, such as paper [16], which analyzes centrality measures indicating the importance of vertices within a graph network and estimates influential individuals or web pages using the PageRank algorithm.

This research develops an end-to-end analytical framework for health insurance fraud detection based on insurance claims data. The framework combines graph theoretic features, community detection, and Random Forest classification to capture both relational and tabular characteristics of insurance claims. Graph-derived attributes and community labels are used

to enrich the feature space and improve fraud detection performance, with all experiments conducted on the original Kaggle[1] dataset.

In this research, a graph theory-based algorithm for health insurance fraud detection is proposed using insurance claims data. The data consist of three datasets. The first dataset includes medical expense records. The second dataset contains details of the expenses. The third dataset includes labels of insured members reviewed and determined by health insurance experts. The data include insured individuals, some of whom are regular members and others identified as fraudulent members.

Each medical bill associated with an insured member is stored as a record in the medical expense dataset, while detailed billing information is stored in the expense details dataset. The dataset utilized in this study was obtained from the publicly available Healthcare Provider Fraud Detection Analysis dataset hosted on the Kaggle platform [9]. This dataset includes inpatient claims, outpatient claims, and beneficiary demographic records, and has been widely used in prior healthcare fraud detection studies.

4 Findings

First, the algorithm will be described. The implementation of the algorithm was carried out using the Python programming language and with the dataset previously mentioned. This dataset includes 138,557 patients, 74,132 doctors, and 5,411 insurance agencies, with a total of 558,211 claims registered for the patients. Each claim includes the start and end date, status, involved physicians (provider, treating physician, operating physician), diagnosis, etc. Patient data includes age, gender, location, race, health conditions, etc.

Important libraries used in this implementation include pandas, infomap, igraph, numpy, seaborn, xgboost, scipy, networkx, etc. The initial part of the implementation involves fetching data from Kaggle, but later, the 3 main data files are merged. These 3 files include patient claims and their relationships, as well as stakeholders. There is also a table of providers or insurance centers to which claims have been referred.

By performing connections, a table is produced that includes patient claims along with their specifications, and for each claim, it is specified whether fraud has occurred or not. Furthermore, by connecting tables, a table of patients, doctors, and the drug provider or insurance center is created. This table includes information about which doctor had what illness related to which claim and which agency reviewed this connection.

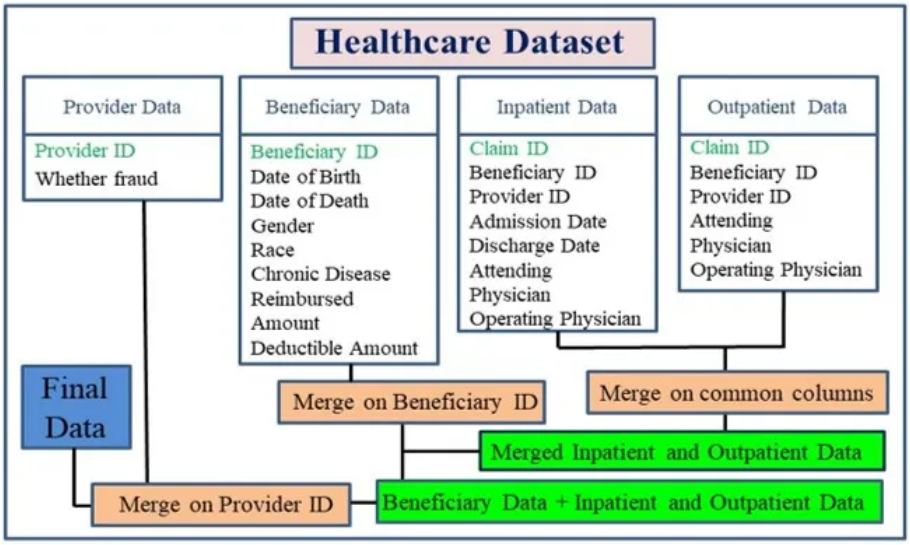
In fact, the following table is formed.

The experiments in this study are conducted on publicly available health insurance claim datasets obtained from the Kaggle platform. The data include (1) inpatient claim records, (2) outpatient claim records, and (3) beneficiary demographic information. Each claim contains details about the patient, the healthcare provider, the attending physician, medical costs,

[1] Kaggle is a widely used online platform for hosting large-scale datasets and machine-learning challenges, providing open access to data for research and reproducible experimentation. Available at: `https://www.kaggle.com`

Table 1. The relationships between patients, doctors, and insurance agencies.

	BeneID	Provider	ClaimID	InscClaimAmtReimbursed	AttendingPhysician	DeductibleAmtPaid	PotentialFraud	Gender	Race	NoOfMonths_PartACov	NoOfMonths_PartBCov
0	BENE100000	1057172	CLM126832	50	2383401	0.0	1	1	1	12	12
1	BENE100000	1057172	CLM351838	70	2370909	0.0	1	1	1	12	12
2	BENE100001	1052145	CLM626521	10	2430032	0.0	0	1	1	12	12
3	BENE100001	1054683	CLM633318	100	2408282	0.0	0	1	1	12	12
4	BENE100001	1054890	CLM332544	90	2343317	0.0	0	1	1	12	12




```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558211 entries, 0 to 558210
Data columns (total 15 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   BeneID                    558211 non-null  object
 1   Provider                  558211 non-null  int64
 2   ClaimID                   558211 non-null  object
 3   InscClaimAmtReimbursed    558211 non-null  int64
 4   AttendingPhysician        558211 non-null  int64
 5   DeductibleAmtPaid         557312 non-null  float64
 6   PotentialFraud            558211 non-null  int64
 7   Gender                    558211 non-null  int64
 8   Race                      558211 non-null  int64
 9   NoOfMonths_PartACov       558211 non-null  int64
 10  NoOfMonths_PartBCov       558211 non-null  int64
 11  IPAnnualReimbursementAmt  558211 non-null  int64
 12  IPAnnualDeductibleAmt     558211 non-null  int64
 13  OPAnnualReimbursementAmt  558211 non-null  int64
 14  OPAnnualDeductibleAmt     558211 non-null  int64
```

Figure 1. Explanation of the data set.

reimbursement amounts, and a binary label indicating whether the claim was identified as potentially fraudulent.

The three datasets were merged using BeneID, Provider, and ClaimID to construct a unified table representing all claim level information. Categorical attributes were normalized to a consistent format, and missing values, primarily in physician identifiers, were handled by standard replacement procedures. The final dataset consists of 558,211 claim records, with 345,415 labeled as non-fraud and 212,796 labeled as fraud.

To prepare the data for downstream analysis, provider and physician IDs were standardized and recoded as numerical identifiers suitable for graph processing. All features were cleaned, type-casted, and aligned to ensure compatibility with both machine learning models and graph index computations.

Next, based on the table formed, a relationship among the agency, the patient, the claim, and the doctor is established. Here, every claim includes a label for fraud or non-fraud.

To analyze structural relationships among key entities involved in the insurance claim process, three interaction networks were constructed: (1) physician-patient, (2) patient-agency, (3) physician-agency.

Each node in these networks represents an entity (patient, physician, or agency), and each edge corresponds to a recorded claim linking the involved parties. For every insurance claim, the dataset specifies the responsible physician, the patient receiving the service, and the agency reviewing the claim. These relationships form a heterogeneous interaction graph from which multiple homogeneous subnetworks can be derived for analytical purposes.

Using the cleaned and merged dataset, an undirected graph G=(V,E) was generated where vertices encode unique entity identifiers, and edges represent co-occurrence within the same claim. Edge multiplicity was collapsed into single connections to simplify centrality computation. The resulting networks capture the underlying relational structure of the claims, enabling the extraction of graph-based behavioral patterns commonly associated with coordinated or organized fraud.

These constructed graphs serve as the foundation for computing topological features such as degree, PageRank, eigenvector centrality, and community structure via the Louvain method, which are subsequently integrated into the machine learning models.

For each claim, there is a doctor responsible for that claim, and thus the information of this doctor is also available. Additionally, for each claim, information about the patient and the insurance agency that reviewed this claim is present. This table serves as the basis for fraud analysis.

Moreover, the table can be utilized to construct a comprehensive network representing the pairwise interactions among the entities. These networks encompass the patient-patient, patient-agency, and physician-agency relationships, as illustrated in the figure below:

In this network, the color of each node is determined based on its degree, i.e., the number of its connecting edges. A table has been constructed that captures the relationships among a claim, a physician, an insurance agent, and a patient. This table also includes patient information and claim attributes, such as whether the claim was fraudulent and whether a payment was made. Subsequently, centrality metrics, namely degree centrality, eigenvector centrality, and PageRank, are calculated for both patients and physicians involved in the network and appended to each corresponding row in the table.

Table 2. Table after adding parameters.

Provider_degree	Provider_eigenvector_centrality	Provider_pagerank	AttendingPhysician_degree	AttendingPhysician_eigenvector_centrality	AttendingPhysician_pagerank
105	0.025261	0.000560	1	0.000821	0.000006
105	0.025261	0.000560	1	0.000821	0.000006
51	0.023621	0.000280	1	0.000768	0.000006
140	0.000059	0.000765	1	0.000002	0.000006
54	0.023700	0.000294	1	0.000771	0.000006

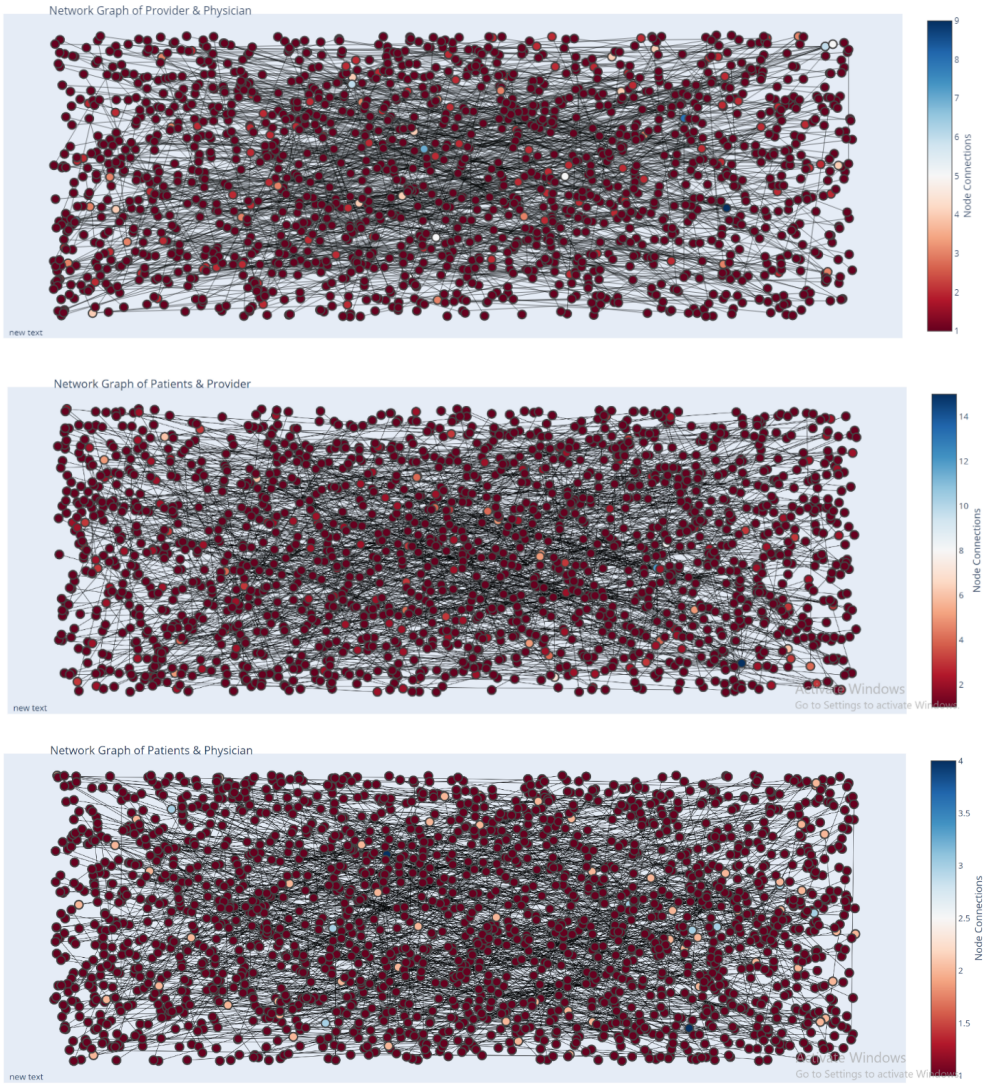


Figure 2. Networks formed.

To better understand how the proposed approach performs, several common machine learning models were also tested on the same dataset. These included Logistic Regression, SVM, Decision Tree, k-Nearest Neighbors, Naive Bayes, Gradient Boosting, AdaBoost, XGBoost, LightGBM, and a simple Multilayer Perceptron.

The results of these baseline models were later compared with the graph-based methods to show how much each step of the analysis improves the overall performance.

After preparing the dataset and adding the network-based features, the learning phase was carried out in three different ways. The goal was to understand how much improvement each step could provide compared with the baseline models introduced earlier.

The first-degree vertex indices are calculated, along with the special features and PageRank for the patient and the related physician, and added to each row of the table. There are 558211 rows that correspond to the claims made, and there are 21 components for each row. In total, there are 345415 non-fraudulent claims and 212796 fraudulent claims. Sixty percent of the data

is used for training and forty percent for testing. Since it is supervised learning, the training data includes the fraud column and is labeled.

Using a Random Forest model, the data is trained in three stages: once with baseline features, once with graph features, and once with graph plus community features. First, learning is performed on 60 percent of the dataset, and then performance is evaluated on the remaining 40 percent. The test set also includes fraud labels to allow evaluation of prediction accuracy. The following steps describe the operation of the Random Forest algorithm:

1. Random samples from a specified training dataset are selected.
2. A decision tree is constructed for each sample.
3. Voting is performed by aggregating the decision trees.
4. The final prediction is determined by majority vote.

For each model, the best parameters and results are examined. Claims are evaluated using three different configurations.

Algorithm 1. Conceptual Workflow of the Proposed Framework

Input: Insurance claims table with patient, physician, agency, and fraud label.
Output: Fraud likelihood for each claim.

1. Construct relational tables linking physician–patient–agency interactions.
2. Build three heterogeneous networks representing their pairwise connections.
3. Compute graph centrality metrics (degree, eigenvector, PageRank) for all nodes.
4. Append graph metrics to the claims table without altering any original fields.
5. Apply Louvain community detection to the physician–agency layer.
6. Add community labels to each claim record.
7. Train Random Forest classifiers under three configurations:
   (a) baseline features only,
   (b) baseline plus graph metrics,
   (c) baseline plus graph metrics plus community labels.
8. Evaluate and compare the three methods using the original dataset.

To ensure a valid evaluation, all steps of data preparation were performed using only the training split. Graph features, including degree-based metrics and PageRank, were computed exclusively from the training portion of the network, and the test data were not used in any way during feature construction or model training. Community detection using the Louvain algorithm was also applied only on the training graph. This separation prevents data leakage and ensures that the reported performance reflects the true generalization ability of the models.

In the first method, the available tabular data containing relationships between patients, doctors, and insurance representatives are used to train machine learning models. In this approach, no graph-based information is included.

In the second method, the same tabular dataset is augmented with graph-based features. Specifically, degree centrality, eigenvector centrality, and PageRank scores are computed for the relevant doctor and insurance representative nodes and appended to each record. In this case, the model achieves an improvement in accuracy from 60% to 90%.

The third and final method, based on the proposed framework, extends the previous approach by incorporating community structure. After computing graph-based features, the Louvain algorithm is applied to detect communities within the doctor–representative network. Following community assignment, Random Forest models are trained within each community on the claim data. This approach further improves accuracy from 90% to 95%.

To isolate the contribution of graph analytics and community structure, an ablation study is conducted with three configurations:

- Machine-learning models using only tabular features,
- Models augmented with graph-structural features, and
- Models using graph features plus Louvain community assignments.

This analysis quantifies the incremental benefits of incorporating network-based information.

Subsequently, feature importance was analyzed and compared. In the context of insurance fraud detection, claims can be further screened and prioritized using these insights. Figure 3, which illustrates the comparative importance of features, supports this analysis. As observed, certain variables such as gender and race have minimal impact on fraud detection. However, insurance agency characteristics are more influential; agencies with a higher number of connections are more likely to be associated with fraudulent claims. Another significant factor is the claim amount, where larger claims exhibit a higher probability of fraud.

Such comparisons assist insurance companies in identifying and focusing on the most influential features. By applying filters based on these key variables, a subset of suspicious claims can be flagged for further investigation. This selective approach enables prioritization of potentially fraudulent claims, improving efficiency and resource allocation.

The reported accuracy values of 60%, 90%, and 95% correspond to the performance of the three evaluated configurations of the proposed framework. These results demonstrate the incremental improvement achieved by incorporating graph-structural features and Louvain-based community information into the classification process.

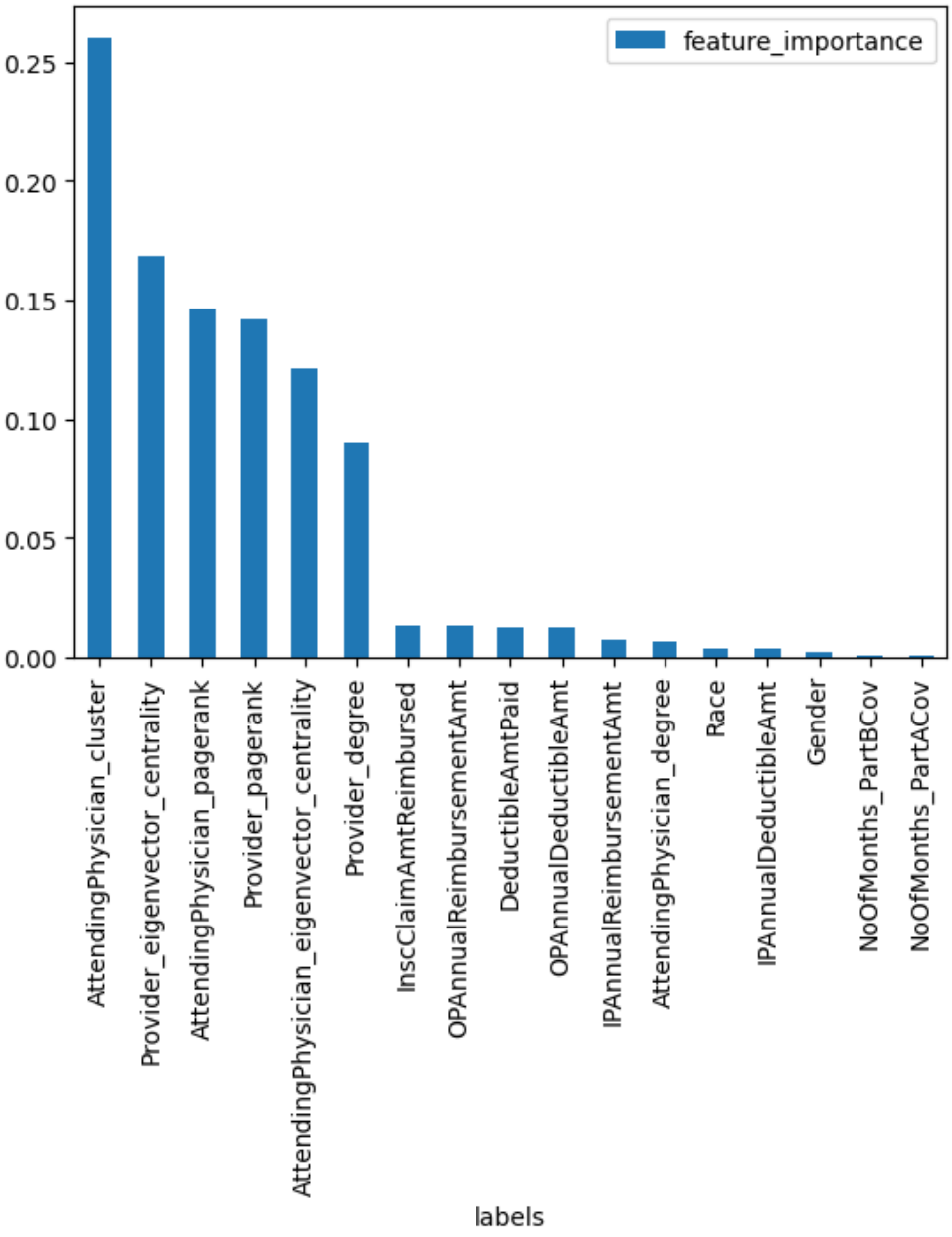


Figure 3. The importance of each variable.

5 Conclusion

Graph theory is a powerful analytical approach that can be utilized for fraud detection in complex relational systems. By modeling interactions among physicians, patients, and insurance agencies as interconnected networks, suspicious behavioral structures can be identified more effectively than with conventional tabular approaches. The use of graph-based analysis enables the discovery of hidden structural relationships that are often difficult to detect using traditional machine learning techniques alone.

The application of graph theory for fraud detection offers several important advantages.

First, it enables the identification of complex interaction patterns that may remain undetected in standard statistical analyses. Second, graph-based approaches can assist in identifying previously unknown fraudulent behaviors and coordinated fraud rings. Third, early detection of suspicious structures can reduce investigation costs and improve the efficiency of insurance monitoring systems.

Despite these advantages, graph-based fraud detection also presents several challenges. The effectiveness of graph analysis depends heavily on the availability of high-quality and well-structured data. In addition, identifying suspicious behavioral patterns within large-scale healthcare networks requires efficient algorithms and substantial computational resources.

In this study, graph theory and community detection algorithms, particularly the Louvain method, were applied to health insurance fraud detection using publicly available datasets obtained from the Kaggle platform. The data included insurance claims, beneficiary information, healthcare providers, and treatment-related costs. First, heterogeneous interaction networks were constructed among physicians, patients, and insurance agencies. Graph-based structural metrics, including degree centrality, eigenvector centrality, and PageRank, were then extracted and incorporated into the classification framework. In the final stage, suspicious communities were identified through the Louvain community detection algorithm and integrated into the predictive model.

The experimental results demonstrated that the inclusion of graph-based structural features significantly improved fraud detection performance. The baseline machine learning approach achieved approximately 60% accuracy, while the addition of graph features increased performance to 90%. Incorporating community detection further improved the overall accuracy to 95%. These findings indicate that relational and structural information embedded within healthcare claim networks can substantially enhance fraud detection capability.

Furthermore, the proposed framework provides a transparent and reproducible analytical workflow that can support future research and practical implementation in healthcare insurance systems. The combination of graph analytics, community detection, and supervised learning offers an interpretable approach for identifying suspicious provider behaviors and potential collusion patterns.

Although the proposed framework achieved promising results, several directions remain for future research. Future studies may explore temporal graph analysis, graph neural networks, and dynamic community detection methods to better capture evolving fraudulent behaviors over time. In addition, integrating explainable artificial intelligence techniques may further improve the interpretability of fraud predictions for healthcare auditors and insurance investigators.

To ensure reproducibility and facilitate future development, the implementation code and supplementary materials have been made publicly available at: `https://github.com/MohammadMehdi41/fraudDetection.git`

Overall, this study demonstrates that integrating graph-based structural analysis and community detection into supervised machine learning pipelines can significantly improve the detection of fraudulent health insurance claims and provide valuable support for intelligent

healthcare fraud monitoring systems.

Funding

This research received no external funding.

Data Availability Statement

Data is contained within the article.

Conflicts of Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] H. Askari Zadeh, M. J. Tarikh, Fraud detection in health insurance based on data mining approach, in: Proceedings of the International Conference on Modern Research in Management, Economics, and the Capability of the Tourism Industry in Development, 2017.
[2] M. Behdad, L. Barone, M. Bennamoun, T. French, Nature-inspired techniques in the context of fraud detection, IEEE Trans. Syst. Man Cybern. C 42 (2012) 1273–1290. https://doi.org/10.1109/TSMCC.2012.2215851
[3] S. Chen, A. Gangopadhyay, A novel approach to uncover health care frauds through spectral analysis, in: IEEE International Conference on Healthcare Informatics (ICHI), IEEE, 2013, pp. 499–504.
[4] S. Chen, A. Gangopadhyay, Health care fraud detection with community detection algorithms, in: 2016 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2016, pp. 1–6. https://doi.org/10.1109/SMARTCOMP.2016.7501694
[5] E. A. Duman, S. Sagiroglu, Health care fraud detection methods and new approaches, in: Proceedings of the International Conference on Computer Science and Engineering, 2017. https://doi.org/10.1109/UBMK.2017.8093544
[6] E. Estrada, P. A. Knight, A first course in network theory, Oxford Univ. Press, Oxford, 2015. https:// api.pageplace.de/ preview/ DT0400.9780191039928 _A24313697/ preview-9780191039928 _A24313697.pdf
[7] FBI, Investigating insurance fraud, 2012. https://www.fbi.gov/ news/ stories/ investigating-insurance-fraud
[8] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174. https:// doi.org/ 10.1016/ j.physrep.2009.11.002
[9] Kaggle, Healthcare provider fraud detection analysis. https://www.kaggle.com/
[10] Y. Li, C. Yan, W. Liu, M. Li, A principal component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification, Appl. Soft Comput. 70 (2018) 1000–1009. https://doi.org/10.1016/j.asoc.2017.07.027
[11] J. Liu, E. Bier, A. Wilson, J. A. Guerra-Gomez, T. Honda, K. Sricharan, L. Gilpin, D. Davies, Graph analysis for detecting fraud, waste, and abuse in healthcare data, AI Mag. 37 (2016) 33–46. https://doi.org/10.1609/aimag.v37i2.2630
[12] A. Mohammadbaghi, Preliminary insurance discussion on third-party insurance, New Developments in the World of Insurance (2005) 25–36.
[13] M. E. J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. U.S.A. 103 (2006) 8577–8582. https://doi.org/10.1073/pnas.0601602103

[14] G. K. Orman, V. Labatut, A comparison of community detection algorithms on artificial networks, in: International Conference on Discovery Science, Springer, Berlin, 2009, pp. 242–256. https://doi.org/10.1007/978-3-642-04747-3_20
[15] N. Rahimian, Fraud detection, Official Accountant (2011) 81–92.
[16] L. Subelj, S. Furlan, M. Bajec, An expert system for detecting automobile insurance fraud using social network analysis, Expert Syst. Appl. 38 (2011) 1039–1052. https:// doi.org/ 10.1016/ j.eswa.2010.07.143

Citation: M. M. Ahmadi, A. Hamzeh, A. Kamandi, Graph-enhanced fraud detection in health insurance: integrating centrality metrics and community analysis for improved accuracy, J. Disc. Math. Appl. 11(2) (2026) 131–151.
https://doi.org/10.22061/jdma.2026.12810.1181

APPENDIX

Implementation Details

The source code provided in this appendix corresponds to the implementation of the proposed end-to-end fraud detection framework described in Sections 3 and 4. The code covers the following components:

1. Data loading and preprocessing of inpatient, outpatient, and beneficiary claim records;
2. Construction of entity interaction graphs among patients, physicians, and insurance agencies;
3. Extraction of graph-based features, including degree, eigenvector centrality, PageRank, and community labels using the Louvain method;
4. Integration of these features into a supervised machine learning pipeline based on a Random Forest classifier with hyper-parameter tuning.

All scripts are intended to ensure reproducibility of the experimental evaluation reported in this study and reflect the computational workflow used to generate the presented results.

```
inpatients = pd.read_csv(
    PATH + 'Train_Inpatientdata-1542865627584.csv'
)
outpatients = pd.read_csv(
    PATH + 'Train_Outpatientdata-1542865627584.csv'
)
beneficiaries = pd.read_csv(
    PATH + 'Train_Beneficiarydata-1542865627584.csv'
)
labels = pd.read_csv(
    PATH + 'Train-1542865627584.csv'
)

df1 = inpatients[
    [
        'BeneID',
        'ClaimID',
        'Provider',
        'InscClaimAmtReimbursed',
        'AttendingPhysician',
        'DeductibleAmtPaid'
    ]
]
df2 = outpatients[
    [
        'BeneID',
        'ClaimID',
        'Provider',
        'InscClaimAmtReimbursed',
        'AttendingPhysician',
        'DeductibleAmtPaid'
    ]
]
df3 = beneficiaries[
    [
        'BeneID',
        'Gender',
        'Race',
        'NoOfMonths_PartACov',
        'NoOfMonths_PartBCov',
```

```
        'IPAnnualReimbursementAmt',
        'IPAnnualDeductibleAmt',
        'OPAnnualReimbursementAmt',
        'OPAnnualDeductibleAmt'
    ]
]

df = pd.concat([df1, df2])

df = (
    df
    .set_index('Provider')
    .join(labels.set_index('Provider'))
    .reset_index()
    .set_index('BeneID')
    .join(df3.set_index('BeneID'))
    .reset_index()
)

df['PotentialFraud'] = (
    df['PotentialFraud']
    .replace("No", 0)
    .replace("Yes", 1)
    .astype(int)
)

df['AttendingPhysician'] \
    .str.removeprefix("PHY") \
    .astype(float) \
    .min()

df['Provider'] = (
    df['Provider']
    .str.removeprefix("PRV")
    .astype(int)
    + 1_000_000
)
df['AttendingPhysician'] = (
    df['AttendingPhysician']
    .str.removeprefix("PHY")
    .fillna(0)
    .astype(int)
    + 2_000_000
)

# defining a dictionary of graph features
nodes_info_dict = {

    # 'closeness_centrality':
    #     nx.closeness_centrality,

    'eigenvector_centrality':
        nx.eigenvector_centrality_numpy,

    'pagerank':
        nx.pagerank
}

columns_with_node_infos = (
    ['degree']
    + list(nodes_info_dict.keys())
)

nodes_info = pd.DataFrame.from_dict(
    dict(nx.degree(G)),
    orient='index'
).rename(
    columns={0: 'degree'}
).reset_index()
```

```
# computing graph features for each node
for info, fun in nodes_info_dict.items():

    temp = pd.DataFrame.from_dict(
        fun(G),
        orient='index'
    ).rename(
        columns={0: info}
    ).reset_index()

    nodes_info = nodes_info.merge(
        temp,
        on='index'
    )


nodes_info = nodes_info.rename(
    columns={'index': 'Physician'}
)

# adding graph features to the dataframe
df_enriched = df.merge(
    nodes_info,
    left_on='Provider',
    right_on='Physician',
    how='left'
).drop(
    'Physician',
    axis=1
)
df_enriched.rename(
    columns={
        k: 'Provider_' + k
        for k in columns_with_node_infos
    },
    inplace=True
)

df_enriched = df_enriched.merge(
    nodes_info,
    left_on='AttendingPhysician',
    right_on='Physician',
    how='left'
).drop(
    'Physician',
    axis=1
)
df_enriched.rename(
    columns={
        k: 'AttendingPhysician_' + k
        for k in columns_with_node_infos
    },
    inplace=True
)

def findCommunities(G):
    """
    Partition network with the Infomap algorithm.
    Annotates nodes with 'community' id and
    return number of communities found.
    """

    im = infomap.Infomap(
        two_level=True,
        silent=True
    )

    print(
        "Building Infomap network "
        "from a NetworkX graph..."
    )
```

```
    for e in G.edges():
        im.addLink(*e)

    print("Find communities with Infomap...")

    im.run()
    print(
        f"Found {im.num_top_modules} modules "
        f"with codelength "
        f"{im.codelength:.8f} bits"
    )
    communities = {}

    for node, module in im.modules:
        communities[node] = module

    nx.set_node_attributes(
        G,
        communities,
        'community'
    )
    return G

X = X.fillna(0)

# Splitting data into train and test sets
X_train, X_test, y_train, y_test = (
    train_test_split(
        X,
        y,
        test_size=0.4,
        random_state=69
    )
)
print(
    f"Shapes after splitting:\n\n"
    f"X_train: {X_train.shape}, "
    f"y_train: {y_train.shape}\n"
    f"X_test: {X_test.shape}, "
    f"y_test: {y_test.shape}"
)

accuracy = []

feature_names = [
    "Baseline",
    "Graph Features",
    "Graph Features with Community Detection"
]

features = [
    # Baseline features
    [
        'InscClaimAmtReimbursed',
        'DeductibleAmtPaid',
        'Gender',
        'Race',
        'NoOfMonths_PartACov',
        'NoOfMonths_PartBCov',
        'IPAnnualReimbursementAmt',
        'IPAnnualDeductibleAmt',
        'OPAnnualReimbursementAmt',
        'OPAnnualDeductibleAmt'
    ],
    # Baseline + Graph Features
    [
        'InscClaimAmtReimbursed',
        'DeductibleAmtPaid',
        'Gender',
```

```
            'Race',
            'NoOfMonths_PartACov',
            'NoOfMonths_PartBCov',
            'IPAnnualReimbursementAmt',
            'IPAnnualDeductibleAmt',
            'OPAnnualReimbursementAmt',
            'OPAnnualDeductibleAmt',
            'Provider_degree',
            # 'Provider_closeness_centrality',
            'Provider_eigenvector_centrality',
            'Provider_pagerank',
            'AttendingPhysician_degree',
            # 'AttendingPhysician_closeness_centrality',
            'AttendingPhysician_eigenvector_centrality',
            'AttendingPhysician_pagerank'
        ],
        # Baseline + Graph Features + Community Detection
        [
            'InscClaimAmtReimbursed',
            'DeductibleAmtPaid',
            'Gender',
            'Race',
            'NoOfMonths_PartACov',
            'NoOfMonths_PartBCov',
            'IPAnnualReimbursementAmt',
            'IPAnnualDeductibleAmt',
            'OPAnnualReimbursementAmt',
            'OPAnnualDeductibleAmt',
            'Provider_degree',
            # 'Provider_closeness_centrality',
            'Provider_eigenvector_centrality',
            'Provider_pagerank',
            'AttendingPhysician_degree',
            # 'AttendingPhysician_closeness_centrality',
            'AttendingPhysician_eigenvector_centrality',
            'AttendingPhysician_pagerank',
            'AttendingPhysician_cluster'
        ]
]

hyper_parameter_grids_RFC = [
        {
            # Grid 1: No regularization
            "randomforestclassifier__criterion":
                ['gini'],
            "randomforestclassifier__max_depth":
                [10, 20, 50, 100, 250, 300, 500],
            "randomforestclassifier__min_samples_split":
                [2, 3, 5, 10, 20, 30],
        },
        {
            # Grid 2: L2 regularization
            "randomforestclassifier__criterion":
                ['entropy'],
            "randomforestclassifier__max_depth":
                [10, 20, 50, 100, 250, 300, 500],
            "randomforestclassifier__min_samples_split":
                [2, 3, 5, 10, 20, 30],
        },
]
```

```
pipeline_RFC = make_pipeline(
    StandardScaler(),
    RandomForestClassifier(random_state=69)
)

for feature in features:
    print("*" * 100)
    print(
        "# Tuning hyper-parameters "
        "for accuracy"
    )
    print("*" * 100)
    print()
    # This performs gridsearch,
    # evaluating each set of
    # hyper-parameters using
    # k-fold cross validation.
    clf = HalvingGridSearchCV(
        pipeline_RFC,
        hyper_parameter_grids_RFC,
        scoring="accuracy",
        cv=4,
        n_jobs=-1
    )
    X_train_subset = X_train[feature]
    X_test_subset = X_test[feature]
    clf.fit(
        X_train_subset,
        y_train
    )
    acc = round(
        clf.best_estimator_.score(
            X_test_subset,
            y_test
        ) * 100,
        2
    )
    print(
        "Best parameters set "
        "found on development set:"
    )
    print(clf.best_params_)
    print(
        "Best score on development set:"
    )
    print(f"Accuracy: {acc}")
    accuracy.append(acc)
```