Identification of Fraudulent Healthcare Claims Using Fuzzy Bipartite Knowledge Graphs

Md Enamul Haque and Mehmet Engin Tozal

***Abstract*—Health insurance is one of the most important services that people depend on for paying the bills related to hospital and clinical services. This dependency on health insurance lures some healthcare service providers to commit insurance frauds which has become a grave concern. The majority of healthcare fraud is committed by a very small number of untrustworthy providers. Yet, such fraudulent actions damage the reputation of the health service providers and cost the system billions of dollars. In this paper, we specifically focus on the fraudulent claim identification problem and develop different solution schemes to identify the fraudulent cases in healthcare claims with minimal data. We present a solution to the fraudulent claim identification problem that translates diagnoses and procedure code's relations into Bipartite Graphs with Fuzzy Edges (`BiGFuzzE`). We also investigate the extension of `BiGFuzzE` using vector representations of clinical codes instead of non-negative matrix factorization (NMF). Our experimental evaluations demonstrate significant outcomes.**

***Index Terms*—Healthcare, Insurance, Fraud, Fuzzy Bipartite Graph, Negative Insurance Claims**

I. Introduction

Many countries face numerous challenges in their public healthcare systems. Among these challenges, medical insurance fraud is of a great importance, as it creates financial loss for both patients and insurance providers. Medical or health insurance fraud is defined as a crime which is committed by a physician or group of physicians within a hospital to gain financial benefits by misrepresenting claims to public or commercial insurance providers [1]. The Center for Medicaid and Medicare Services (CMS) in the U.S. projects that the national healthcare expenditure (NHE) will increase to nearly $6 trillion by 2027 [2]. A recent study shows that Medicare and Medicaid paid an estimated $68 billion in improper payments to the service providers that include fraud, waste, and abuse [3]. The estimated financial losses due to the fraudulent and fabricated claim submission is in the orders of tens of billions of dollars in the United States [4]. However, an insignificant portion of the total fraudulent cases are captured and recovered despite the strict policies regarding fraud and abuse [5].

Typically, a small number of dishonest providers commit health insurance frauds, which include false diagnoses to validate medically unnecessary procedures; billing for more expensive procedures than the actually performed; fabrication of claims; and unbundling of procedures into multiple smaller procedures. Furthermore, misrepresenting the need for procedures that are not covered by the insurance companies such as plastic surgeries, is also a method to obtain financial benefits by the fraudulent providers. Nonetheless, due to the volume, variety, and velocity of the data, it is not practical to apply only domain knowledge to identify these frauds. Data mining techniques are effective ways to analyze big data sources to detect fraudulent claims at an early stage.

In this paper, we focus on the problem of healthcare fraud detection from the perspective of health insurance providers which include both private and government organizations. Our prime goal is to identify unusual relations between diagnosis-procedure pairs within a claim where limited but standardized data is available. Our solution involves only diagnosis and procedure codes which are common to medical claims among other information such as patient data and claim amount. Existing methods try to identify frauds in insurance domain that corresponds to different sources such as healthcare providers, policy holders, and insurance amounts. In most of the cases, these methods are applied on private datasets. In addition, the features used in those datasets are diverse and generally not compatible with each other to transfer a solution approach to different healthcare software systems. Furthermore, the health insurance companies are more reluctant to share the data that involves patient specific information compared to other sectors, such as retail and distribution. Therefore, we limit our problem formulation to diagnosis and procedure codes within a claim which can be processed similarly without restricting to a country or healthcare management software. Moreover, these medical codes are created and managed by international organizations or national institutions. The limited data usage also supports the Healthcare Insurance Portability and Accountability Act (HIPAA) in the US, the General Data Protection Regulation (GDPR) in Europe, or similar law in other parts of the world. Our solution approach assumes that the claim data contains fuzzy relationships between diagnosis and procedure code pairs in International Classification of Diseases (ICD) coding format. The proposed approach also works on other coding formats such as Current Procedural Terminology (CPT), Healthcare Common Procedure Coding System (HCPCS), or their combinations.

Table I presents a sample treatment claim consisting of three diagnosis and three procedure codes. Note that the insurance claims do not convey the descriptions of the medical codes. We obtained the descriptions from an online catalog to enrich the information presented in the table [6].

Md Enamul Haque is with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, 70503 USA e-mail: enamul@louisiana.edu.

Mehmet Engin Tozal is with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, 70503 USA e-mail: metozal@louisiana.edu.

https://doi.org/10.1109/TSC.2023.3296782

TABLE I
A SAMPLE POSITIVE CLAIM DATA CONTAINING ICD-9 DIAGNOSIS AND PROCEDURE CODES.

Diagnosis Code	ICD9v3 Description
41401	Coronary atherosclerosis of native coronary artery
4019	Other diagnostic procedures on lymphatic structures
V4581	Aortocoronary bypass status
Procedure Code	**ICD9v3 Description**
3722	Left heart cardiac catheterization
42789	Other specified cardiac dysrhythmias
25000	Incision of tendon sheath

We represent the historical claim data as a fuzzy bipartite graph where the nodes represent ICD, CPT, and/or HCPCS format diagnosis and procedure codes. A fuzzy bipartite graph is a graph such that the nodes are divided into two disjoint sets D and P where no two vertices from the same set share an edge between them. In addition, the edge weights are bounded between 0 and 1, representing the normalized occurrance frequency or probability of two codes appearing together in a claim. More formally, a fuzzy bipartite graph G is defined as $G =< D, P, E >$, where $D = \{d_i | 1 \leq i \leq m\}$, $P = \{p_j | 1 \leq j \leq n\}$, $E \subseteq D \times P$, and $0 \leq E_{ij} \leq 1$. We present an example fuzzy bipartite graph in Figure 1, where D and P contain diagnosis and procedure codes, respectively. The diagnosis set D and procedure set P contain clinical codes representing different diagnoses and procedures such as 7366 ("other acquired deformities of knee") and 311 ("temporary tracheostomy"). The edge weight, E_{ij}, between a diagnosis-procedure pair denotes the probability of the pair appearing in a claim within the historical claims. For instance, diagnosis code 0414 and procedure code 7301 has 0.01 probability of appearing in a claim together.

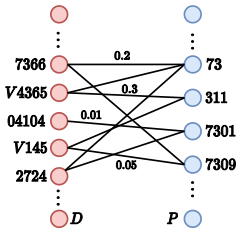


Fig. 1. A partial fuzzy bipartite graph representation of a claim consisting of diagnosis and procedure codes using fuzzy edges.

We construct Bipartite Graph with Fuzzy Edges (`BiGFuzzE`) from the verified and reimbursed historical insurance claims. Our goal in representing the claims using a fuzzy bipartite graph is to be able to evaluate the strength of direct relations between diagnoses and procedures. Moreover, it helps us to assess the strength of indirect relations between diagnoses and procedures through diagnosis to procedure paths. Next, we use non-negative matrix factorization on the reliable path length matrix to generate diagnoses' and procedures' feature components. The path lengths allow us to primarily identify the relevancy of a procedure and diagnosis within a test claim. The diagnosis component facilitates computing the similarity among diagnoses codes which is employed in deciding a procedure as fraudulent or legitimate in the final step of `BiGFuzzE`.

Our unique contributions in this study are summarized as follows.

- We formulate the fraudulent claim identification problem using fuzzy bipartite graphs and matrix factorization over standardized claim data consisting of diagnosis and procedure codes.
- We utilize reliable path lengths on the bipartite knowledge graph to evaluate the relevancy between a pair of codes.
- We introduce clinical code-level similarity and compatibility within a claim to improve fraud identification tasks.

We compare `BiGFuzzE` with its diagnosis and procedure component generation and neighbor selection methods based on vector embeddings. In our default `BiGFuzzE`, we use non-negative matrix factorization on the reliable path length matrix for diagnosis and procedure component generation. We initialize the components using vector representations of clinical codes in `BiGFuzzE+vector` approach. We employ a random selection classifier that labels a claim as either fraudulent or non-fraudulent as a baseline comparison. Similar to [7], we represent a claim as a sum of vector representations of ICD9 and CPT codes. Then, we use the vector representations as input to a traditional RNN with unidirectional LSTM [8]. We also implement a bi-directinal RNN [9] to efficiently represent the clinical codes in the vector space for comparison. Additionally, we compute relevance scores among diagnoses by non-negative matrix factorization and cosine similarity to support the fraud identification procedure. One important aspect of the proposed approach is that it can provide procedure specific fraud scores for diagnoses, which is not possible using the present methods (to the best of our knowledge).

We collect health insurance claim dataset from the Center for Medicaid and Medicare Services (CMS) which includes verified and disbursed claims with diagnosis and procedure codes. The dataset contains inpatient and outpatient claims from years between 2008-2010. An inpatient visit refers to services provided to patients with overnight stay in a hospital or clinic by a physicians' written order. On the other hand, outpatient visits include emergency medical service, X-rays, or any other services that do not require doctors' written order. Our experimental results show that Bipartite Graph with Fuzzy Edges (`BiGFuzzE`) reaches an accuracy, precision, and recall scores of 95%, 100%, and 87%, respectively on the inpatient dataset acquired from CMS. Additionally, it demonstrates 97%, 100%, and 94% accuracy, precision, and recall scores, respectively on the outpatient dataset. The proposed method will directly benefit the health insurance companies and federal government services such as Medicaid and Medicare in the U.S. and abroad.

The rest of the paper is organized as follows. Section II presents healthcare fraud related studies in general, including both claim and provider level anomalies. Section III presents a formal description of our problem setup for fraudulent claim identification. Section V discusses detailed experimental results. Finally, we conclude the paper in Section VI.

II. Related Work

According to the National Health Care Anti-Fraud Association (NHCAA), healthcare fraud is defined as a deliberate act of deceit or distortion of information that is submitted to organizations such as insurance companies with being aware that the act would result financial gain [10]. Several works in the literature propose solutions to the problem of fraudulent claim identification and misuse of government provided insurance in medical, pharmaceutical, and related domains. However, most of the approaches are limited in terms of autonomous data analysis, which performs independent fraud identification from massive claim data.

Recent approaches to find fraudulent claims include fraud detection in Medicare data without the provider labels by applying machine learning models [11]. Bauder et al. [12]–[14] proposed a fraud detection system that is similar to identifying outliers in large payment systems using multiple predictors. They applied their method on Medicare payment and observed promising results [15]. They also used provider specialty and payment distribution data to identify fraudulent providers [16]. However, their assumption on the fraudulent claims belonging to the providers from exclusion list is impractical as the decision does not consider global claim information and disease-procedure relations.

Mosley and Kucera used clustering, association analysis, and principal component analysis of RIDIT [17, 18] scores to find fraud in medical claims [19]. Similarly, Yang and Hwang [10] built a healthcare fraud detection model with the help of clinical pathways. Their method also demonstrated hand crafted features for differentiating various types of claims. They applied their methods on a real-world data set collected from National Health Insurance (NHI) program in Taiwan. Although the authors constructed different features to generate patterns for both normal and abusive claims, they did not mention the significance of those features.

Settipalli et al. [20] employed multivariate analysis and a weighted multi-tree representation of claims to analyze healthcare data. Their primary emphasis was on provider-specific characteristics to detect potential instances of fraud. However, their approach is not well-suited for identifying fraud at the claim level. In contrast, our approach enables claim-level testing even in situations where provider data is unavailable, allowing us to uncover claim-level fraud more effectively.

Farbmacher et al. [21] introduced text analysis to incorporate unstructured data into healthcare insurance fraud detection. This approach successfully distinguished between fraudulent and non-fraudulent claims, enabling accurate payment disbursement. Although our current solution relies on structured data, we can enhance our method by incorporating unstructured data as well, following the approach described in their study.

Haque et al. [22] showcased the effectiveness of detecting health insurance claim fraud by utilizing a combination of clinical concepts extracted from insurance claims, such as diagnosis and procedure codes. While our approach to problem-solving differs considerably from their study, we still leverage similar structured data obtained from the claims to develop our solution.

Zhang et al. [23] proposed a Medicare fraud detection framework using anomaly detection method. The proposed method consists of spatial density based algorithm. The authors show that the method is more suitable compared to simple local outlier factor (LoF) when applied on medical insurance data. To overcome the issue of LoF, Kose et al. [24] used interactive machine learning to incorporate knowledge base in the unsupervised learning algorithms to identify fraudulent and suspicious cases in healthcare domain.

Wang and Luo [25] presented a probabilistic programming model based on Stan [26] to build an enhanced Beta regression model and applied on Neuroprotection Exploratory Trials in Parkinson's Disease (PD) study. Bauder and Khoshgoftaar [16] proposed a general outlier detection model using Bayesian inference implemented in Stan [27] to screen healthcare claims. Sowah et al. [28] used Genetic Support Vector Machines (GSVMs) to demonstrate improved fraud detection and classification performance, resulting in reduced processing time and increased accuracy compared to traditional SVM classifiers on health insurance claims. Unlike solutions based on structured healthcare claims, Mackey et al. [29] and Alnuaimi et al. [30] created a healthcare fraud and abuse blockchain framework and prototype on the Ethereum platform, utilizing essential blockchain tools and layers such as consensus algorithms, smart contracts, tokens, and governance based on digital identity.

In general, we observe that the problem of detecting frauds in medical domain has been identified using different approaches such as classification methods, Bayesian analysis, statistical surveys, and expert analysis. Different data mining approaches such as natural language processing, social network analysis, text mining, temporal analysis, graphs, and higher order feature constructions are used to discover knowledge from massive healthcare data as well [31]–[34]. In addition, many studies use private data or hand-picked non-standard features for fraud detection. As a result, transferring these solutions to other systems or directly comparing these approaches turn into a challenge by itself.

Unlike most other approaches, our solution approach represents the claim data (diagnosis-procedure relations) using a fuzzy bipartite graph termed as knowledge graph, where diagnoses and procedure codes are considered as the prime elements of a claim. Under this setting we tackle the problem of flagging a procedure as legitimate or fraudulent using neighborhood based similarity of diagnoses for a particular procedure code within a test claim.

III. Fraudulent Claim Identification Process

In this section, we formally introduce the fraudulent healthcare claim identification problem and its solution using Bipartite Graphs with Fuzzy Edges (`BiGFuzzE`). We execute our solution approach on publicly available health insurance claims data. In the following, we first describe a brief background on bipartite graphs with fuzzy edges. Next, we formally present the problem and solution by describing the

process to create a bipartite knowledge graph from the health insurance claims. Finally, we demonstrate the overall solution approach using *offline* pre-computation and *online* fraudulent claim identification transformations and algorithms. Moreover, we describe individual components of the transformations and algorithms using illustrative examples.

A. Fundamental Definitions

Fuzzy graphs are usually defined in two categories where either both vertices and edges or only the edges are considered fuzzy. Rosenfield defined a fuzzy graph where both vertices and edges follow a membership function as:

Definition 1. "A fuzzy graph $G = (\sigma, \mu)$ consists of pair of functions σ and μ where $\sigma : S \to [0, 1]$ and $\mu : S \times S \to [0, 1]$ for all $x, y \in S$ where $\mu(x, y) \leq \sigma(x) \wedge \sigma(y)$." [35]

On the other hand, Yeh et al. defined a fuzzy graph where only edges follow a membership function as:

Definition 2. "A fuzzy graph $G_f = (V, R)$ where V is a set of vertices and R is fuzzy relation on V in which the edges connecting the vertices in V have membership function $\mu_R : V \times V \to [0, 1]$." [36]

We represent the healthcare claims as a bipartite graph where diagnoses and procedures are kept in two separate sets. Additionally, we choose the membership function to assign edge weights to the bipartite graph by following Definition 2.

Definition 3. A fuzzy bipartite graph $\mathcal{B}_f = \langle D, P, E \rangle$ where D and P are two distinct sets of vertices and E denotes the edges connecting the vertices between D and P, where $E_{ij} \to [0, 1] \forall d_i \in D, p_j \in P$.

B. Problem Statement

We assume that our dataset consists of healthcare claims with diagnoses/procedure code information. The i-th claim consists of a set of diagnoses and a set of procedures codes. The claim data from healthcare providers is viewed as a bipartite graph $G = \langle D, P, E \rangle$, where $D = \{d_i : 1 \leq i \leq m\}$, $P = \{p_j : 1 \leq j \leq n\}$, and $E \subseteq D \times P$. D and P are referred as diagnosis and procedure code sets, respectively. E refers to the fuzzy edge set representing the immediate relations between D and P. We assume the graph G has m diagnosis in set D and n procedures in set P. We present our fraud identification algorithms using a bipartite graph where a fuzzy edge weight refers to the probability of a pair of diagnosis and procedure codes to appear in a claim together. Figure 2 is an example fuzzy bipartite graph with four diagnoses and four procedures. The weights of each link is computed using the normalized occurrance frequency (probability) of a diagnosis-procedure pair appearing in a claim in our dataset. The corresponding bi-adjacency matrix of Figure 2 is shown in Figure 3 where each row and column represents a diagnosis and a procedure, respectively. Note that, we use the bi-adjacency matrix in our solution approach as a knowledge graph representing the immediate relations between diagnoses and procedures.

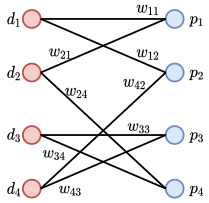


Fig. 2. An illustrative bipartite graph representing relations between diagnoses and procedures.

$$\mathcal{B} = \begin{pmatrix} w_{11} & w_{12} & 0 & 0 \\ w_{21} & 0 & 0 & w_{24} \\ 0 & 0 & w_{33} & w_{34} \\ 0 & w_{42} & w_{43} & 0 \end{pmatrix}$$

Fig. 3. Bi-adjacency matrix representation of the bipartite graph shown in Figure 2 where w_{ij} denotes that diagnosis i and procedure j are linked $w_{ij}\%$ of times compared to all other occurrences in the historical positive claims. Each row and column correspond to a diagnosis and procedure, respectively.

Our problem formulation asks: given a ground truth knowledge graph $\mathcal{B}_f$, how can we flag procedure p_i as legitimate or fraudulent for a new test claim $\mathcal{C}_t$, assuming that the ground truth data contains all historical claims. Note that, the bipartite knowledge graph is a one time computation and can be updated with the introduction of new diagnosis and procedure codes in the claims. A new test claim data $\mathcal{C}_t$ is represented as a diagnosis-procedure vector tuple $(\{d_1, d_2, \ldots, d_k\}\{p_1, p_2, \ldots, p_l\})$, where d_i and p_j refer to the i-th diagnosis and j-th procedure, respectively. To give a simple example, consider a new test claim $\mathcal{C}_t$ containing tuples $(\{d_1, d_4\}\{p_1, p_3, p_4, p_5\})$.

IV. Problem Solution

In this section, we present `BiGFuzzE` using a twofold approach which includes *offline* and *online* computation processes. The *offline* computation process involves building a bipartite knowledge graph. The *online* computation process involves verifying a new claim against the pre-built knowledge graph. First, we demonstrate the *offline* computation elements with comprehensive illustrations and discussions. Next, we explain the *online* computation steps using an algorithm, an illustrative example, and discussions. Finally, we present the interpretation of our fraudulent claim identification process using a sample claim.

A. Offline Pre-computation Steps

We demonstrate the offline computation steps in Figure 4. Our primary goal during these computations is to build the knowledge graph, $\mathcal{B}_f$, and all pairs diagnoses similarity score, $\mathcal{S}$, to use as inputs for the *online* computation process later.

Fuzzy Bipartite Graph: We construct a fuzzy bipartite graph $\mathcal{B}_f$ using the probability between a diagnosis and procedure pair from the claims dataset. The diagnosis and procedure components are extracted from the dataset to be used as two disjoint sets in the bipartite graph. The edge weights, E_{ij} between diagnosis d_i and procedure p_j is computed using the

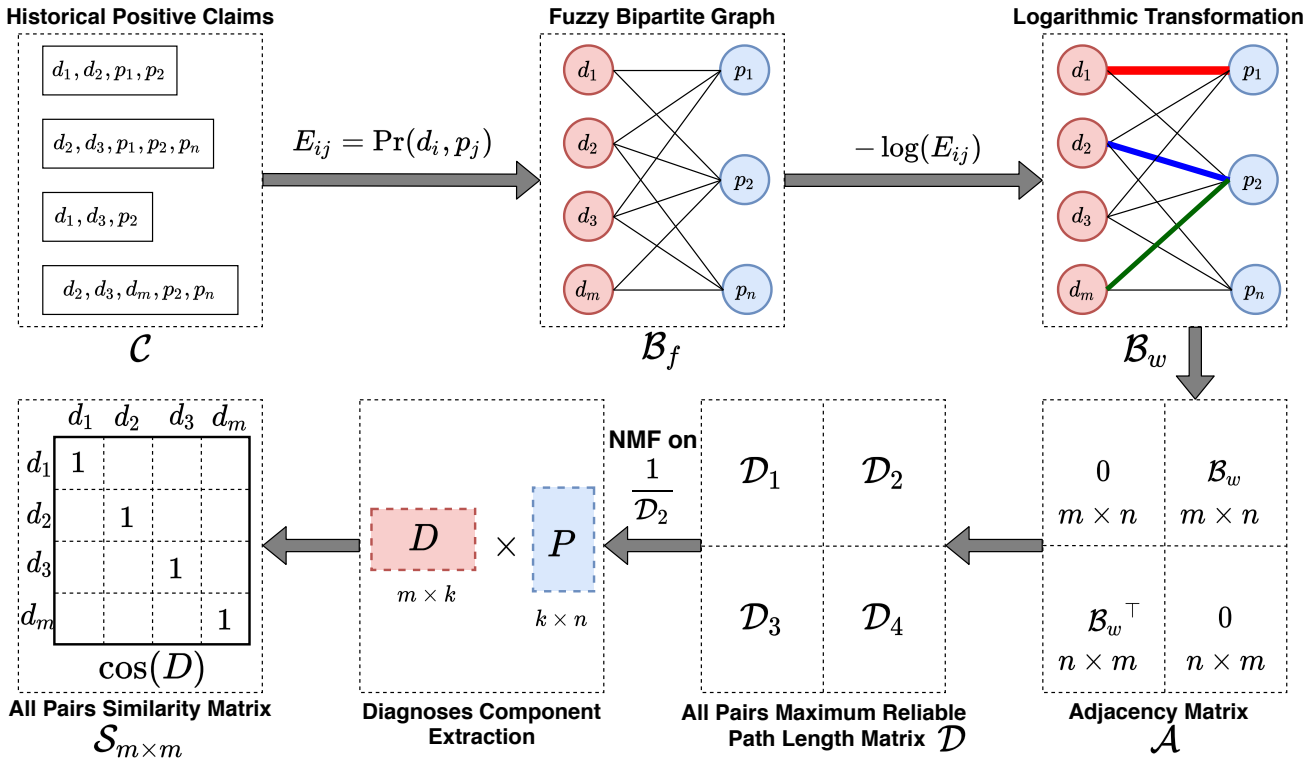


Fig. 4. Offline computation processes to demonstrate individual steps to store fuzzy biaprtite graph $\mathcal{B}_f$ and similarity matrix $\mathcal{S}$ that store bipartite graph structure of the historical positive claims and all pairs diagnoses similarity scores. Higher width of the edge in $\mathcal{B}_w$ denote high reliable path.

probability $\Pr(d_i, p_j)$. In a bipartite graph, edges exist between vertices of two different types, providing various levels of proximity between vertex pairs. Following the modeling of first-order proximity [37, 38], we model explicit relations between a pair of vertices in the fuzzy bipartite graph. The probability of a pair of diagnosis and procedure appearing in a claim is defined using Equation 1.

$$\Pr(d_i, p_j) = \frac{w_{ij}}{\sum\limits_{e_{ij} \in E} w_{ij}} \tag{1}$$

where w_{ij} denotes the number of times d_i and p_j appeared together in a claim. e_{ij} refers to a link between the i-th diagnosis and j-th procedure code in the bipartite graph $\mathcal{B}_f$. The edge weights between a diagnosis and procedure falls between 0 and 1 and stores the pairwise significance of diagnosis and procedure pairs. If a diagnosis and procedure pair is very frequent in the dataset, it will have a higher probability.

Definition 4. First-order (local) proximity: The first order proximity of a bipartite graph $G = \langle V_1, V_2, E \rangle$ is the local pairwise distance between nodes from two vertex sets. For every pair of vertices from set V_1 and V_2 that are connected via an edge e_{ij}, the frequency of that edge w_{ij} denotes the first order proximity between i and j where $i \in V_1$ and $j \in V_2$. The proximity is non-zero only if there is an edge between the target node to others.

Logarithmic Transformation: The *offline* computation steps require shortest path computation between all pairs of diagnoses and procedures in $\mathcal{B}_f$. The shortest path lengths are used as an input to the matrix factorization algorithm to generate diagnosis and procedure components, which facilitates diagnosis and procedure similarity computations. As the edge weights of the bipartite knowledge graph are represented using probabilities, the direct application of the shortest path algorithm is not appropriate. If we directly apply all-pairs shortest path algorithm such as Johnsons algorithm on the fuzzy bipartite graph, the paths with higher probability scores may assume less importance. However, in our problem setting, we consider an edge as important when the probability is relatively higher. We illustrate the problem of direct application of all-pairs shortest path on an example bipartite graph in Figure 2. We redraw Figure 2 in Figure 5 for better understanding and clarity with example edge weights.

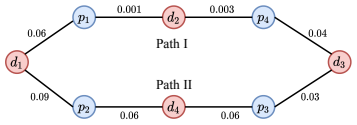


Fig. 5. Redrawn layout of Figure 2 to illustrate logarithmic edge transformation for shortest path computation.

Suppose that we want to compute the shortest path between nodes d_1 and p_4. The shortest path algorithm have two paths to choose from d_1 to p_4. For the first path that includes nodes d_1, p_1, d_2, and p_4, the total weight is 0.064. The second path with nodes d_1, p_2, d_4, p_3, d_3 and p_4 has total weight 0.279.

Therefore the first path will be selected as the shortest path from d_1 to p_4. However, if the algorithm selects the first path we are giving more priority to weak links such as p_1 to d_2 and d_2 to p_4 having probabilities 0.001 and 0.003, respectively. On the other hand, if we use negative logarithm transformation on the edges of the fuzzy bipartite graph before applying all pairs shortest path algorithm, we get 15.53 and 14.76 for the first and second paths, respectively. In this case, the second path is selected rightfully.

By following the maximum reliability path [39, 40], the edges of the fuzzy bipartite graph, $\mathcal{B}_f$ is transformed using a negative logarithmic transformation which will allow us to apply Johnson's all-pairs shortest path algorithm on the fuzzy bipartite graph. We transform the fuzzy bipartite graph $\mathcal{B}_f$ to $\mathcal{B}_w$ so that the edge $E_{\mathcal{B}_w} = -\log\big(E_{\mathcal{B}_f}\big)$ as shown in Figure 4. This subproblem is formulated as follows. Suppose that we are given a bipartite graph $\mathcal{B}_w = \langle D \cup P, E\rangle$ on which each edge $(d_i, p_j) \in E$ has an associated probability score $\Pr(d_i, p_j)$ that represents the reliability of a diagnosis-procedure relation from vertex d_i to vertex p_j. In order to compute the most reliable path between a pair of vertices within the graph, our goal is to find a path $\hat{r}$ such that the product of the edge probabilities within that path is maximized. Let d_i be the source diagnosis and p_j be the destination procedure, and let R be the set of all possible paths from d_i to p_j. The maximum reliable path $\hat{r}$ [39] is then defined using Equation 2.

$$\hat{r} = \arg\max_{r \in R} \prod_{(d_k, p_l) \in r} \Pr(d_k, p_l) \tag{2}$$

where r denotes a path from source d_i to destination p_j and represented as $r = (d_i, p_*), \ldots, (d_k, p_l), \ldots, (d_*, p_j)$. The problem in Equation 2 is transformed into a shortest path problem by translating the edge weights using negative logarithm. As the logarithm is a strictly monotonically increasing function, we can apply the logarithm on Equation 2 without changing the maximization problem using Equation 3 [39].

$$\begin{aligned} \hat{r} &= \arg\max_{r \in R} \left(\log \prod_{(d_k, p_l) \in r} \Pr(d_k, p_l) \right) \\ &= \arg\max_{r \in R} \sum_{(d_k, p_l) \in r} \log \Pr(d_k, p_l) \end{aligned} \tag{3}$$

Next, we can transform the maximization problem in Equation 3 into the minimization probelm in Equation 4 by negating the sum and keeping the objective unchanged [39]. This transformation helps us to apply the shortest path algorithm on the edge weights $(-\log(r_i))$ of the bipartite graph.

$$\begin{aligned} \hat{r} &= \arg\min_{r \in R} \left(- \sum_{(d_k, p_l) \in r} \log \Pr(d_k, p_l) \right) \\ &= \arg\min_{r \in R} \sum_{(d_k, p_l) \in r} - \log \Pr(d_k, p_l) \end{aligned} \tag{4}$$

As our bipartite graph is very sparse, we apply Johnson's algorithm to solve the all pairs shortest path problem on the transformed graph. The initialization of edge probabilities takes $O(E)$ time, and the rest is the same as Johnson's algorithm. Therefore, the algorithm runs in $O\big(VE \log V\big) + O(E)$ or $O(VE \log V)$ where $V = |D + P|$, which is equivalent time complexity of Johnson's algorithm.

Adjacency Matrix and Shortest Paths: As we are dealing with a bipartite graph, the shortest path algorithm can not be directly applied on the bi-adjacency matrix. Therefore, we transform the bi-adjacency matrix into adjacency matrix $\mathcal{A}$ as illustrated in Figure 4.

Definition 5. All pairs shortest paths (APSP) of a graph refers to finding the shortest path between all pairs of vertices in the graph and it corresponds to the maximum reliability path in our setup. Formally, let $G = (V, E)$ be a graph consisting a vertex set V and edge set E. For each pair of nodes (u, v) in V, a subset of edges from E need to be selected so that the subset contains a simple path with start and end vertex of u and v and has minimal weighted sum. If there exists no such path the shortest path set is empty and its length is infinite.

All-pairs maximum reliability path matrix $\mathcal{D}$ is computed from an adjacency matrix. We employ Johnson's [41] all-pairs shortest path algorithm on the bi-adjacency matrix. The algorithm requires a square adjacency matrix where we represent every pair of vertices with their edge weights. However, bipartite graphs do not have inter-connection within the nodes of individual sets, hence the bi-adjacency matrix does not capture all pairs relationship. Therefore, we transform the initial bi-adjacency matrix $\mathcal{B}_w$ into a sparse adjacency matrix $\mathcal{A}$ using the following matrix operation in Equation 5.

$$\mathcal{A} = \begin{pmatrix} \mathbf{0} & \mathcal{B}_w \\ \mathcal{B}_w^\top & \mathbf{0} \end{pmatrix} \tag{5}$$

Considering the bi-adjacency matrix with rows as diagnoses and columns as procedures, we get the following four blocks in the final all-pairs maximum reliable paths matrix, $\mathcal{D}$.

$$\mathcal{D} = \left[\begin{array}{c|c} \mathbf{D} \rightsquigarrow \mathbf{D}_{m \times m} & \mathbf{D} \rightsquigarrow \mathbf{P}_{m \times n} \\ \hline \mathbf{P} \rightsquigarrow \mathbf{D}_{n \times m} & \mathbf{P} \rightsquigarrow \mathbf{P}_{n \times n} \end{array} \right] \tag{6}$$

As we only require the path lengths between diagnosis-procedure pairs, we can use either the top-right or bottom-left block from matrix $\mathcal{D}$. In our experiments, we used the top right block. We compute the *closeness* or relevancy score between all pairs of diagnosis-procedure codes using the reciprocal of the maximum reliable path lengths from the bipartite graph, $\mathcal{B}_w$. The *closeness matrix* $\mathcal{Z}$ is computed from the second component of matrix $\mathcal{D}$ using Equation 7.

$$\mathcal{Z}_{ij} = \begin{cases} \frac{1}{|d_i, p_j|}, & \text{if } d_i \rightsquigarrow p_j \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $d_i \rightsquigarrow p_j$ refers to a valid path between diagnosis d_i and procedure p_j, and $|d_i, p_j|$ denotes the maximum reliable path-length between them.

Diagnoses Similarity: To address the subproblem, *finding similarity between a pair of diagnoses*, we apply a two-step process: i) latent factor modeling on the closeness matrix to get

diagnosis and procedure components and ii) diagnosis similarity matrix using cosine metric from the diagnosis component of the latent factor modeling.

Given a row vector (diagnosis component in the latent space) $d_i \in \mathbf{D}$, we need to compute a similarity score for each row node $d_j \in \mathbf{D}$ where $i \neq j$. The final outcome is a 1-by-m vector consisting of all the similarity scores from d_i where m refers to the number of unique diagnoses. We assume that the diagnoses and procedures share a common latent space where they can be factorized into two separate components each describing the properties of diagnoses and procedures, respectively. To materialize the assumption, we consider non-negative matrix factorization on the *closeness* matrix and get two new components for diagnoses and procedures in the latent space. Let the *closeness* matrix $\mathcal{Z}$ share a latent space by two non-negative matrix components $\mathbf{D}$ and $\mathbf{P}$ such that:

$$\mathcal{Z} \approx \mathbf{DP} \tag{8}$$

where $\mathbf{D}$ and $\mathbf{P}$ denote the latent space representations of diagnosis and procedure codes. We decompose the matrix $\mathcal{Z}$ into two components with dimensions $m \times k$ and $k \times n$, respectively where $k < \min(m, n)$. m, n, and k denote the number of diagnoses, procedures, and latent factors, respectively.
To find an approximate factorization of $\mathcal{Z} \approx \mathbf{DP}$, we first define a cost function that quantify the quality of approximation by minimizing the distance between $\mathcal{Z}$ and the matrix product $\mathbf{DP}$. We used Frobenius norm as the distance function which is an extension of the Euclidean norm to matrices [42]. Therefore, the optimization problem for decomposing the closeness matrix into diagnosis and procedure components is formulated using Equation 9.

$$\arg\min_{D,P>0} \frac{1}{2}||\mathcal{Z} - \mathbf{DP}||_F^2 \tag{9}$$

NMF is an NP-hard Problem and its complexity is $\mathcal{O}(mnr)$ on most real-world problems [43], where m, n and r are the number of rows, columns and the desired rank of a matrix.

After we decompose the closeness matrix $\mathcal{Z}$ into $\mathbf{D}$ and $\mathbf{P}$, we construct the similarity matrix $\mathcal{S}$ for all diagnosis pairs by employing the cosine similarity metric using Equation 10. Note that the cosine similarity metric allows us to ignore the magnitude of the diagnoses in the latent factor space, but consider their dimensions.

$$\mathcal{S}(d_i, d_j) = \begin{cases} \frac{\sum_{k=1}^{r} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{r} {d_{ik}}^2} \sqrt{\sum_{k=1}^{r} {d_{jk}}^2}}, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{10}$$

where d_i and d_j are the rows of $\mathbf{D}$, representing the diagnoses in the latent space. The similarity matrix $\mathcal{S}$ is symmetric since the edges of the bipartite graph in our problem formulation are undirected.

B. Online computation Process

In this part, we present the *online* computation process to classify new claims using our proposed algorithm and an illustrative example.

Online Algorithm: We demonstrate the working principles of the fraudulent claim identification process using an algorithmic template presented in Algorithm 1. The algorithm expects a test claim $\mathcal{C}_t$, the fuzzy bipartite knowledge graph $\mathcal{B}_f$, a decision threshold λ, and the diagnoses similarity matrix $\mathcal{S}$ as inputs, and returns a test claim status Ψ, verifying whether the claim is fraudulent or not. The decision threshold λ is a hyper-parameter of our algorithm which is initialized between 0.01 and 0.5. This can be set appropriately based on the system requirements. Higher values of λ enforces strict rules for fraudulent claim identification. On the other hand, lower values of λ can produce high false positives. Note that, both the knowledge graph and the similarity matrices are precomputed during the offline process, which contributes to lower computational complexity in the verification process.

Lines 1-3 initialize the test claim status, global similarity, and new procedure link frequency. The claim staus Ψ is initialized to false, indicating non-fraudulent. Global similarity and procedure link frequencies are set to zero. Global similarity is used to store the total similarity score for all new daignosis-procedure pairs within the test claim. Line 3 stores the number of procedures $\mathcal{L}$ that have complete new assignment within a test claim. For example, if a procedure p_i have at least one new shared edge with the diagnosis group, then $\mathcal{L}$ is increased by one.

Definition 6. Trust set $\mathcal{R}$ denotes a set of diagnoses codes, which have links to a procedure in the knowledge graph and has non-zero probability. On the other hand, non-trust set $\mathcal{R}'$ contains all diagnoses from the test claim which are not included in the trust set $\mathcal{R}$. Therefore $\mathcal{R}' = \{D_t\} \setminus \mathcal{R}$ where $\{D_t\}$ is the diagnoses set of the test claim.

At lines 4-27, Algorithm 1 traverses all the procedures within the test claim $\mathcal{C}_t$ to compute the local and global similarity scores. Line 5 initializes the local similarity score σ_l. Line 6 initializes the number of links α between a procedure and the diagnoses group, which were not seen previously in the historical claims. Local similarity score is computed for every procedure with respect to the diagnoses set within the test claim. For example, if a test claim consists of m diagnoses $\{d_1, d_2, \ldots, d_m\}$ and one procedure, the local similarity score will be computed in m iterations. At each iteration, a diagnosis will be compared with the trust set of the procedure using diagnoses similarity matrix $\mathcal{S}$. If the procedure is linked with trusted diagnoses set $\mathcal{R}$ which do not overlap with the m diagnoses within the test claim, then the value of the number of unseen links α will be m and the local similarity σ_l for a single procedure within the test claim is computed using Equation 11.

$$\sigma_l = \frac{\sum_{i=1}^{\alpha} \sum_{j=1}^{|\mathcal{R}|} \mathcal{S}(d_i, \rho_j)}{|\mathcal{R}| * \alpha} \tag{11}$$

where ρ_j denotes j-th diagnosis within the trust set $\mathcal{R}$. This computation is performed at line 22.

Line 7 initializes a binary variable Φ to ensure the use of a procedure only once with the group of diagnoses in trust

Algorithm 1: Online algorithmic framework for fraudulent claim identification.

Input: New test claim, $\mathcal{C}_t = \{D_t, P_t\}$
Input: Fuzzy bipartite graph $\mathcal{B}_f$ // precomputed knowledge graph from the offline process
Input: All pairs diagnoses similarity matrix $\mathcal{S}$ // precomputed from the offline process
Input: Decision threshold similarity values λ
Output: Test claim status, Ψ

1: Test claim status $\Psi \leftarrow$ False // test claim status is initialized to non-fraudulent
/* Online computation steps for fraudulent score */
2: Global similarity $\sigma_g \leftarrow 0$
3: New procedure links $\mathcal{L} \leftarrow 0$ // number of procedures with at least one new link
4: **for** $\forall p_i \in P_t$ of $\mathcal{C}_t$ **do**
5: Local similarity $\sigma_l \leftarrow 0$
6: Temporary active links $\alpha \leftarrow 0$
7: $\Phi \leftarrow$ False
8: **for** $\forall d_j \in D_t$ of $\mathcal{C}_t$ **do**
9: **if** $\mathcal{B}_f[p_i][d_j] \leq 0$ **then**
10: $\alpha \leftarrow \alpha + 1$
11: **if** $\Phi ==$ False **then**
12: $\mathcal{L} \leftarrow \mathcal{L} + 1$
13: $\Phi \leftarrow$ True
14: $\mathcal{R} \leftarrow$ find diagnoses set from p_i using $\mathcal{B}_f$ // diagnoses stored in trust set with respect to p_i
15: **end if**
16: **for** $\rho \in \mathcal{R}$ **do**
17: $\sigma_l \leftarrow \sigma_l + \mathcal{S}(\rho, d_j)$ // similarity scores between a pair of diagnoses
18: **end for**
19: **end if**
20: **end for**
21: **if** $\alpha \neq 0$ **then**
22: $\sigma_l \leftarrow \frac{\sigma_l}{|\mathcal{R}| * \alpha}$ // local similarity scores between a p_i and other diagnosis
23: **else**
24: $\sigma_l \leftarrow 1$ // when all the relations have non-zero probability in $\mathcal{B}_f$
25: **end if**
26: $\sigma_g \leftarrow \sigma_g + \sigma_l$ // accumulating global similarity from local similarities
27: **end for**
28: **if** $\mathcal{L} > 0$ **then**
29: $\sigma_g \leftarrow \frac{\sigma_g}{\mathcal{L}}$ // average global similarity
30: **if** $\sigma_g \leq \lambda$ **then**
31: $\Psi \leftarrow$ True // mark $\mathcal{C}_t$ as fraud for additional verification
32: **end if**
33: **end if**
34: **return** Ψ

set $\mathcal{R}$ if at least one link is new. The algorithm proceeds by comparing this variable on line 11 to prohibit locating the same trust set for multiple times. Lines 8-20 traverses the diagnoses of claim $\mathcal{C}_t$ for computing the local similarity σ_l for a single procedure. Line 9 checks whether a link between a procedure and a diagnosis is valid from the bipartite graph $\mathcal{B}_f$. Line 10 increments the temporary active links variable that is required to keep track of the number of non-existent diagnosis-procedure links within the test claim. Line 11 verifies the indicator variable Φ once a link between a procedure and a diagnosis is not found in $\mathcal{B}_f$. This also ensures that the procedure link $\mathcal{L}$ is updated only once for a procedure from the test claim if the procedure has at least one new diagnosis. Line 13 resets the indicator variable for a procedure to avoid multiple increments of $\mathcal{L}$. It also looks up for the trust set only once during each iteration to avoid computational complexity at line 14. Lines 16-18 finds similarity scores between a diagnosis from the test claim and the diagnoses set which share positive edge weights with the procedure.

Lines 21-25 computes local similarity score σ_l if at least one new link is found, otherwise the link is already in $\mathcal{B}_f$. Line 26 sums up all local similarity scores to get the total global similarity score for all procedures within claim $\mathcal{C}_t$. Line 29 computes the average global similarity score using the new procedure link count $\mathcal{L}$ in Equation 12.

$$\sigma_g = \frac{\sum_{p_i \in P_t} \sigma_l^{p_i}}{\mathcal{L}} \tag{12}$$

where p_i is a member of the procedures set P_t from test claim C_t and $\sigma_l^{p_i}$ denotes local similarity of procedure p_i from Equation 11. Line 31 assigns the claim status label based on the threshold value λ, which is chosen empirically. Note that, the value of λ should be chosen according to the level of flexibility that can be tolerated by the organization that is going to use the system for the claim reconciliation purposes. Finally, line 36 returns the test claim status Ψ.

Algorithm 1 requires a precomputed knowledge graph $\mathcal{B}_f$ and a diagnoses similarity matrix $\mathcal{S}$. The memory complexity of our approach is $|D||P| + |D||D| = |D|(|P| + |D|)$ where $|D|$ is the number of diagnoses and $|P|$ is the number of procedures with respect to a medical coding system. The algorithm runs in $|P_t||D_t||D|$ time, which is asymptotically bounded by $|P||D||D|$ where $|D|$ is the number of diagnoses and $|P|$ is the number of procedures with respect to a medical coding system.

Once we apply our fuzzy bipartite graph method on a new sample claim, we flag the claim based on the pre-computed closeness and similarity scores from the ground truth knowledge graph. In the following, we present an illustrative example to briefly visualize the working principles of the online algorithm presented in Algorithm 1.

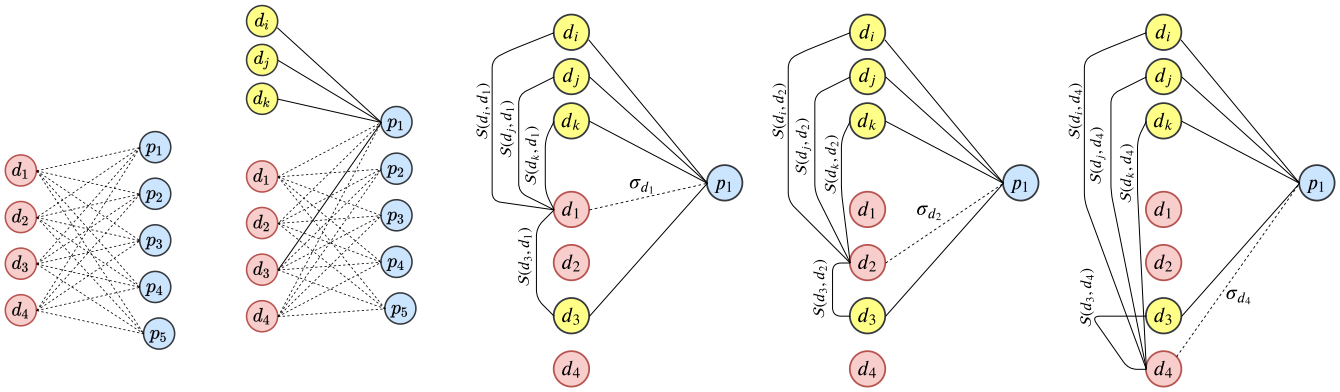
Illustrative Example: Figure 6 presents a simplified example computing the fraud score for a procedure within a test claim. Although, the example test claim consists of five procedures, we only demonstrate the online algorithm applied on a single procedure as the rest follow the same process.

Let us assume that a test claim $\mathcal{C}_t$ consists of four diagnoses $\{d_1, d_2, d_3, d_4\}$ and five procedures $\{p_1, p_2, p_3, p_4, p_5\}$. Our goal is to identify whether the claim $\mathcal{C}_t$ is fraudulent or not. We explain the figure using the fraud score computation for procedure p_1. The remaining procedures are treated similarly to compute the overall fraud score for $\mathcal{C}_t$. In Figure 6(a) the claim with four diagnoses and five procedures are presented with a fully connected bipartite graph. We represent the connection between every diagnosis and procedure pair using dashed lines to emphasize that the edges (or relations) need to be verified. In Figure 6(b) we demonstrate the relation between a diagnosis and procedure using both dashed and solid edges. The dashed edges denote a complete bipartite graph between all pairs of diagnoses and procedures in $\mathcal{C}_t$, similar to Figure 6(a). The set of diagnoses that are connected with a procedure using solid edges denote the trusted diagnoses set $\mathcal{R}$ for that procedure which is extracted from the pre-computed knowledge graph. Note that, a diagnosis from a test claim, which is initially marked as dashed, can also be in the trusted set for a procedure after analyzing the knowledge graph. It means that the specific diagnosis-procedure pair in the test claim is valid. In our example, diagnosis d_3 belongs to both the test claim and the trust set of procedure p_1. Therefore the edge between p_1 and d_3 are legitimate. We find the diagnoses from the test claim which are not present in the trusted set $\mathcal{R}$ for a particular procedure. Next, we compute the similarity scores between a diagnosis of non-trust set $\mathcal{R}'$ and all the members of $\mathcal{R}$. We repeat this process for all non-trust set diagnoses and the procedure. For instance, the diagnoses members of non-trust set are d_1, d_2, and d_4 for procedure p_1 as shown with dashed lines.

In Figure 6(c), we demonstrate the first iteration of the online algorithm where the similarity lookup between a non-trust diagnosis d_1 and all members of the trust set $\{d_i, d_j, d_k, d_3\}$ is computed. The goal of this operation is to validate the relation between diagnosis d_1 and procedure p_1 with respect to the knowledge graph $\mathcal{B}_f$. The similarity score σ_{d_1} is computed by averaging the lookup similarity scores from the precomputed similarity matrix $\mathcal{S}$, mentioned at line 17 of Algorithm 1. Similar to the first iteration, we demonstrate the second and third iterations for procedure p_1 in Figures 6(d) and 6(e), respectively. In the second iteration, the similarity scores are looked up from $\mathcal{S}$ for non-trust diagnosis d_2 and all members of the trust set members $\{d_i, d_j, d_k, d_3\}$ using $\mathcal{S}(d_2, d_i), \mathcal{S}(d_2, d_j), \mathcal{S}(d_2, d_k)$, and $\mathcal{S}(d_2, d_3)$, respectively. In the third iteration, we compute σ_{d_4} by using the same process. Therefore, for a procedure within the test claim every iteration computes the average similarity score of the corresponding diagnosis with the existing diagnoses. If the average score is very low, the diagnosis within the test set is likely to be a fraud and needs further investigation. In our example, the overall similarity score with respect to procedure p_1 is further averaged with respect to the number of non-trusted diagnoses as $\sigma_{p_1} = (\sigma_{d_1} + \sigma_{d_2} + \sigma_{d_4})/(|\mathcal{R}| * |\mathcal{R}'|)$.We follow the similar process for the remaining procedures p_2, p_3, p_4, and p_5 of the test claim. The average similarity score is computed with respect to all the five procedures of test claim $\mathcal{C}_t$ as $\sigma_g = (\sigma_{p_1} + \sigma_{p_2} + \cdots + \sigma_{p_5})/|P_t|$. Finally, a fraudulent flag is assigned to claim c_t if the similarity score is less than a predefined threshold parameter λ.

C. Interpretation

Interpretable models are one of the prime requirements in the healthcare domain for predictive tasks, such as claim status identification. The model should be explainable so that the theory behind the algorithm is well understood. In `BiGFuzzE`, we achieve this by analyzing the *closeness* matrix and neighborhood selection steps. Closeness matrix stores the relevancy between all pairs of diagnosis and procedures codes and later used to generate diagnosis components in the latent space. On the other hand, the trust set $\mathcal{R}$ finds a set of diagnoses that are directly related to a procedure p_j within a test claim. When the procedure p_j is tested for suspicion with respect to a valid diagnosis code d_i, first the procedure code p_j is searched in the knowledge graph $\mathcal{B}_f$ where it is in the direct neighborhood of the diagnosis d_i. If there is no link available in $\mathcal{B}_f$, we then search for the similarity between d_i and the trusted set of p_j including the diagnoses from $\mathcal{B}_f$ that are immediately connected to p_j via positive edge weights. Our assumption is that if the procedure is not directly related to the diagnosis d_i, the neighbor diagnoses, d_k, of the procedure might be related to the diagnosis d_i, where $k \neq i$ and $k \in \{1, 2, \ldots, m\}$. Finally, the procedure p_j is marked



(a) Test claim (b) Test claim setup (c) First iteration (d) Second iteration (e) Third iteration

Fig. 6. Illustration of the online computation process for procedure p_1. Dashed edges refer to the relations between diagnoses and procedure codes which needs verification. Solid edges denote the trust set $\mathcal{R}$ from the knowledge graph $\mathcal{B}_f$.

as suspicious if the average similarity score between the test diagnoses and the trust set $\mathcal{R}$ goes below predefined decision threshold λ.

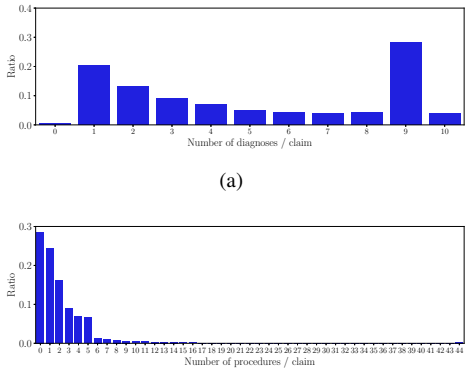
V. Empirical Evaluations

We conducted experiments on two claims data to determine if BiGFuzzE offers significant prediction performance compared to the baseline and RNN based models. Note that, we do not compare our approach with the existing fraud detection studies that use CMS claim data, because all the methods use non-clinical code based features such as provider payment information and physicians' previous historical claim amounts with different problem formulation [11]–[16]. In addition, our clinical-code based synthetic negative claim generation methods are not applicable for those studies. To the best of our knowledge, there is no existing study that benefits from using diagnoses and procedure codes within the claim while identifying claim-based frauds. As the diagnoses and procedure codes not only contain rich information regarding treatments but also remain unchanged due to universal coding format within a claim, we only emphasize on these codes in our developed BiGFuzzE models.

In this section, we first describe the experimental setup followed by the evaluations of BiGFuzzE with alternative and baseline models. Then, we present a qualitative visualization of the diagnoses and procedures relations that demonstrate the fraudulent claims.

A. Datasets

We collect health insurance claim dataset from the Center for Medicaid and Medicare Services (CMS) which includes verified and disbursed claims with diagnosis and procedure codes. The dataset contains inpatient and outpatient claims between years 2008-2010. The claims include medical diagnosis and procedure codes in conjunction with de-identified zip code, beneficiary payments, and patient information. In our BiGFuzzE model, we discard this information, as not only it limits our model to certain software or healthcare provider but also it is irrelevant to our problem definition. We apply *offline* computation on the historical positive claims from two random data files from the CMS dataset. We perform two sets of experiments in the *online* testing process to demonstrate the efficacy of BiGFuzzE using 33,387 *inpatient* and 39,540 *outpatient* positive claims from two randomly selected data files. The *offline* and *online* computation data do not overlap. Note that, our test data includes corresponding synthetic negative claims as well.



(a)

(b)

Fig. 7. Diagnosis and procedure frequency distributions per claim.

Figure 7 summarizes the combined inpatient and outpatient positive claims with respect to the number of diagnosis and procedures within a claim. As shown in the figures both distributions have long tails with a spike at nine in Figure 7(a), because doctors often make zero to a few diagnoses and apply zero to several procedures as part of a treatment.

Synthetic negative claim generation: One important challenge in evaluating the fraud identification models is the lack of negative ground truth claims. We generate synthetic negative claims based on the bipartite relations between diagnosis and procedure pairs within historical positive claims [44]. The first step in our negative claim generation method is to generate probabilities over all procedures in the ground truth positive claims. We use *minimum pooling* and *softmax* procedure probability assignment to produce such probabilities. Finally, we randomly draw negative procedures according

to the computed probability distribution to generate negative claims by replacing the procedures in a set of positive claims.

Given a candidate negative claim $\mathcal{C} = \{D_i, P_i\}$, let $D_i = \{d_1, d_2, \ldots, d_k\}$ have corresponding distance vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$. Each row, $\mathbf{v}_j$ of second block of the distance matrix $\mathcal{D}$ in Equation 6 denotes the relevancy between diagnosis d_j and all procedures. Minimum pooling aggregates all distance vectors into a single vector $\mathbf{v}$ by computing element-wise minimum of the distance vectors as shown in Equation 13.

$$\mathbf{v}_i = \min\{\mathbf{v}_{1i}, \mathbf{v}_{2i}, \ldots, \mathbf{v}_{ki}\} \tag{13}$$

where $\mathbf{v}_{ji}$ denotes the distance of the j-th diagnosis to the i-th procedure and $\mathbf{v}_i$ is the distance of the diagnosis group to the i-th procedure. We use minimum pooling approach in the negative claim generation process because the distance of a group of diagnoses to a particular procedure should not be more than the closest diagnosis to the procedure.

Next, given a pooled diagnosis-procedure distance vector $\mathbf{v}$, we assign probabilities to procedures using the *softmax* function in Equation 14, which translates the distance vector $\mathbf{v}$ into probability vector $\mathbf{u}$ over all procedures.

$$\mathbf{u}_i = \begin{cases} 0 & \text{if } \mathbf{v}_i = 1 \\ \dfrac{\exp\left(\frac{1}{\mathbf{v}_i}\right)}{\sum_{j=1}^{n} \exp\left(\frac{1}{\mathbf{v}_j}\right)} & \text{otherwise} \end{cases} \tag{14}$$

where n denotes the number of procedures in the original bipartite graph $\mathcal{B}_f$. We replace a procedure of a positive claim based on randomly selected procedure from P according to the probability distribution $\mathbf{u}$.

B. Experimental Results

In this part, first we present the experimental design that includes Recurrent Neural Network (RNN) and baseline methods for comparison. We compare two versions of `BiGFuzzE` with these methods. Note that, we use Skip-gram [45] model to transform the ICD codes into vector embeddings, as the RNN method requires code embedding as input. We ran our experiments on a server machine with 48 Intel Xeon CPUs and 512 GB RAM, running the Ubuntu flavor of Linux OS.

The first two proposed models include `BiGFuzzE` and `BiGFuzzE+vector`. The other models include RNN+uni, RNN+bi, and baseline. Our model has one hyper-parameter λ that controls the flexibility of the fraud detection. In addition, we have procedure replacement probability parameter that is used for generating different levels of negative claims.

`BiGFuzzE`: During the knowledge graph generation phase, input claims are transformed into a fuzzy bipartite graph $\mathcal{B}_f$. Then $\mathcal{B}_f$ is passed to logarithmic transformation and adjacency matrix computation. Next, the adjacency matrix is fed to the Johnson's all-pairs shortest path algorithm to compute the closeness matrix $\mathcal{Z}$ based on the logarithmic graph transformation. The closeness matrix $\mathcal{Z}$ is later factored into diagnosis and procedure components to compute similarity scores between diagnosis pairs. The matrix factorization step using *non-negative matrix factorization* method requires the *factor* size as a parameter which we empirically chose to be 20 as other choices such as 50 and 100 produce similar outcomes.

`BiGFuzzE+vector`: We use a similar setup as `BiGFuzzE`, except the diagnoses and procedures are initialized according to the vector representations of the medical codes. We apply Skip-gram [45] model on the historical claim data to produce the vector representations.

Baseline: As a naive baseline, we employ a random selection classifier that labels a claim as either fraudulent or non-fraudulent. This approach explains the effect of diagnosis neighborhoods of procedures.

RNN+uni: Similar to [7], we use traditional RNN with a uni-directional LSTM [8]. The model computes average of ICD9 and CPT code embeddings to represent claim embeddings and uses that as inputs to the model.

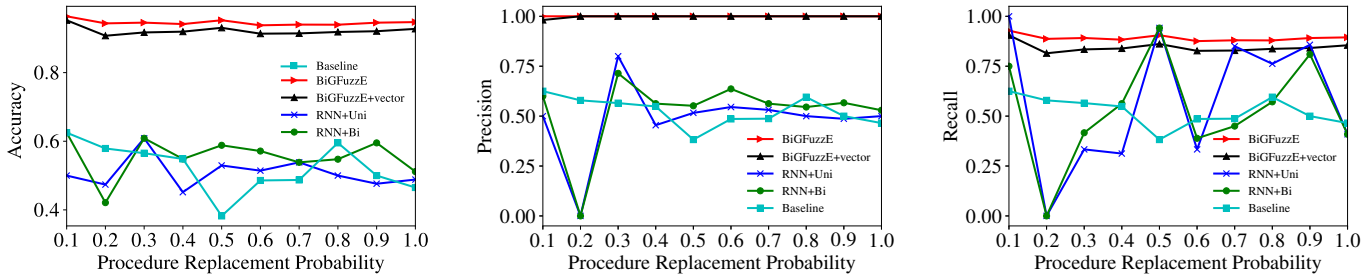
RNN+bi [9]: It is similar to RNN+uni in terms of using recurrent neural network and clinical code features. However, we apply a bidirectional LSTM [46] to efficiently embed the clinical codes in the vector space in RNN+bi.

TABLE II
A SAMPLE POSITIVE CLAIM DATA CONTAINING ICD-9 DIAGNOSIS AND PROCEDURE CODES.

Diagnosis code	**ICD9 description**
7366	Other acquired deformities of knee
V4365	Knee joint replacement
04104	Streptococcus infection
V145	Personal history of allergy to narcotic agent
2724	Other and unspecified hyperlipidemia
2720	Pure hypercholesterolemia
53081	Esophageal reflux
V5866	Long-term (current) use of aspirin
4254	Other primary cardiomyopathies
Procedure code	**ICD9 description**
73	Other procedures inducing or assisting delivery
311	Temporary tracheostomy

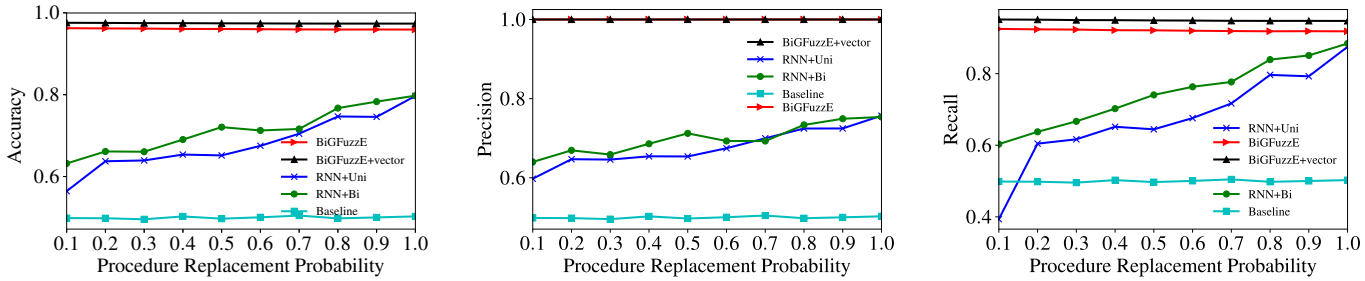
The primary goal of our experiments is to predict whether a claim consists of fraudulent procedures with respect to its diagnoses set. To substantiate the goal, we conduct two sets of experiments and demonstrate the effectiveness of `BiGFuzzE` using two different datasets: inpatient and outpatient claims. A patient is categorized as inpatient if the hospital stay is longer and prescribed by authorized doctors for relevant procedures. On the other hand, a patient is categorized as an outpatient if he/she gets lab test, X-rays, or any other hospital services without the written order from a doctor to admit to a hospital as an inpatient. Each claim includes a set of diagnoses and procedures among many other attributes such as de-identified patient information. We demonstrate a sample positive claim containing ICD-9 (International Classification of Diseases 9th Revision) diagnosis and procedure codes in Table II. Note that, the CMS dataset only includes the codes in the claim, the code descriptions are retrieved from an online catalogue to augment the codes [6].

Our proposed method, `BiGFuzzE` evaluates test claims that include positive claims similar to Table II and synthetic negative claims generated by the process introduced in Section V-A. We measure accuracy, precision, and recall scores



(a) Accuracy (b) Precision (c) Recall

Fig. 8. Results on the inpatient dataset with respect to replacement probability threshold



(a) Accuracy (b) Precision (c) Recall

Fig. 9. Results on the outpatient dataset with respect to replacement probability threshold

on claim datasets with varying levels of negative procedures controlled by procedure replacement probability parameter. The larger the replacement probability, the higher the average number of irrelevant procedures in negative claims.

Effect of Procedure Replacement Probability: In this part, we present accuracy, precision, and recall scores of our methods and the compared models with respect to inpatient and outpatient datasets. Note that, we present average results for all the methods using ten procedure replacement probability thresholds to demonstrate the impact of the variation of negative claims on the results. In addition, we use vector length 20 for claim embedding for both RNN+uni and RNN+bi approaches.

Figure 8 presents accuracy, precision, and recall scores for all methods with respect to varying probability threshold on inpatient claim data. The accuracy scores in Figure 8(a) shows that `BiGFuzzE` and `BiGFuzzE+vector` approaches achieve average scores between 91% and 95%. `BiGFuzzE` uses non-negative matrix factorization (NMF) on the closeness matrix for diagnosis component generation. On the other hand, `BiGFuzzE+vector` uses medical code embeddings generated using the *Skip-gram* model. The matrix factorization based `BiGFuzzE` encodes the distance between a pair of codes from the historical positive claims through closeness matrix. The medical code embeddings encode the relationship of codes with respect to their neighboring codes within a claim. Note that the inpatient claims are smaller in terms of the number of clinical codes. Therefore, unlike NMF based embeddings on the closeness matrix, the vector embedding has limited capability to capture the relationship among the codes. As a result, we observe that `BiGFuzzE` performs relatively better compared to `BiGFuzzE+vector` considering the accuracy scores.

We observe that RNN with unidirectional and bidirectional LSTM and Baseline approaches achieve accuracy scores between 0.41 and 0.62 on varying procedure probability thresholds. Since RNN based methods use code embeddings to represent a claim as features to classify as either non-fraudulent or fraudulent, the performance is not promising. Smaller number of codes within the inpatient claims also affect the results. A claim is represented as the average embeddings of all the codes that make up the claim. In addition, both negative and positive claims share the exact diagnoses codes in our problem formulation. The only difference in the negative claims is the procedure codes which are controlled by the procedure replacement threshold. The code embeddings capture the similarity of diagnoses or procedure codes with respect to the most commonly paired neighboring codes within a claim. As a result there exists insignificant differences in the code embeddings between positive and negative claims. Therefore, we observe similar results when RNN based methods are used for classification compared to the random Baseline method which randomly assigns class labels to a claim.

The precision scores in Figure 8(b) also demonstrate a similar trend for all methods. `BiGFuzzE` and `BiGFuzzE+vector` demonstrate high precision with respect to replacement threshold of 0.2 and higher. The results show that `BiGFuzzE` and `BiGFuzzE+vector` have insignificant false positives. RNN and Baseline methods perform poorly because of the same reason of lower accuracy scores. The recall scores in

Figure 8(c) demonstrate inconsistent behavior for RNN and Baseline approaches. BiGFuzzE and BiGFuzzE+vector demonstrate consistent results with minimal false negatives. Overall, BiGFuzzE and BiGFuzzE+vector demonstrate high accuracy, precision, and recall scores with minimum false negatives for inpatient claims. Note that, the false negatives will add additional human intervention to verify a positive claim which was labeled as negative during the testing phase.

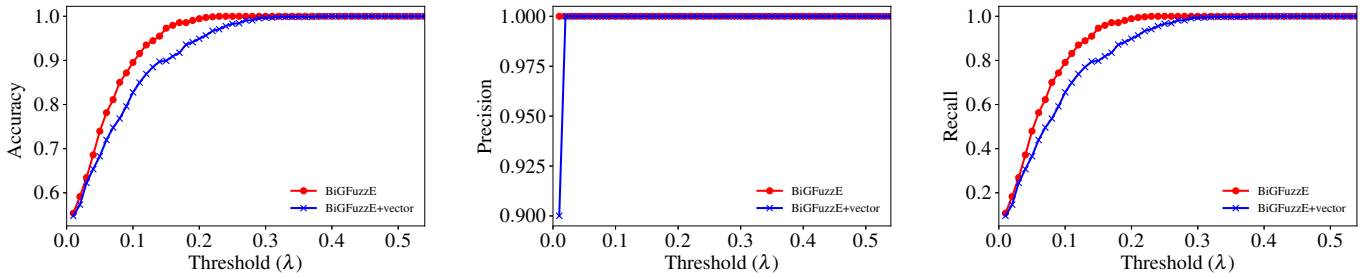
Figure 9 presents the accuracy, precision, and recall scores of all methods with varying procedure replacement thresholds on the outpatient data. Note that, outpatient claims generally consist of more procedures compared to the inpatient claims. An inpatient claim consists of maximum of ten and five diagnoses and procedures codes, respectively. On the other hand, an outpatient claim has maximum of ten and forty-four diagnoses and procedures, respectively. The difference between inpatient and outpatient claims in terms of the number of procedures is very critical. Because, difference not only influences the evaluation scores but also explains the reason behind such evaluations.

Figure 9(a) presents accuracy scores for all methods. BiGFuzzE and BiGFuzzE+vector outperform consistently over all the replacement thresholds and demonstrate scores between 95% and 97%. We also notice that BiGFuzzE+vector performs better compared to BiGFuzzE. The reason is that the code embeddings can make a distinguishable feature set between positive and negative claims due to the numbers of higher procedures. The test claims are identified with high precision as our methods construct knowledge graph on the positive claims. The results demonstrated in Figure 9(b) also suggest that BiGFuzzE and BiGFuzzE+vector can accurately detect negative claims. However, we observe that BiGFuzzE records insignificant false negatives which contribute to a relatively lower recall in Figure 9(c). The scores for RNN based classifier shows improvements compared to the results of inpatient claims as the number of procedure codes are higher in outpatient clams. However, the baseline approach shows approximately scores around 50% due to the random label assignment on the test claims.

Parameter sensitivity analysis: In this part, we demonstrate the sensitivity of accuracy, precision, and recall scores with respect to the decision threshold parameter λ on both inpatient and outpatient datasets in Figures 10 and 11, respectively. We initialized the decision parameter $\lambda = \{0.01, 0.02, 0.03, \ldots, 0.55\}$. We observe that the results on both datasets are better when λ is 0.3 or more. The decision threshold parameter in our algorithm is used to find the similarity between a trust set $\mathcal{R}$ and all diagnoses within a claim. If the overall similarity is lower than the threshold then the algorithm labels the claim as fraudulent.

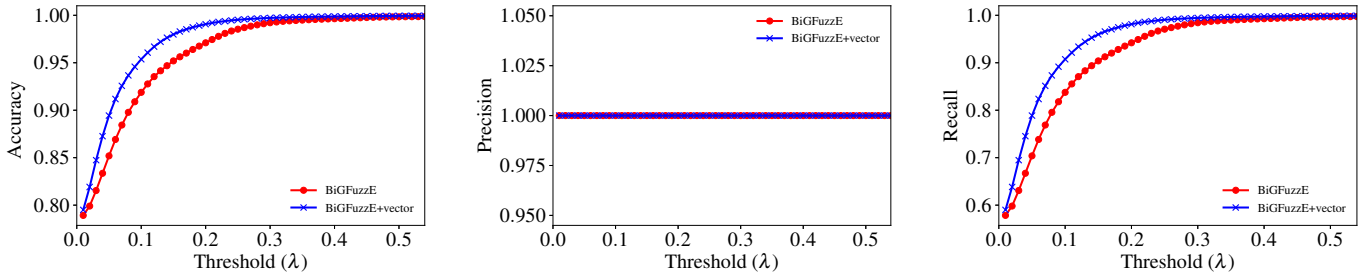
Figure 10 presents accuracy, precision, and recall scores based on different decision threshold, λ, applied on inpatient dataset. We explain the results based on the components used to represent diagnosis similarity in our algorithm. We prepare the diagnosis and procedure components using non-negative matrix factorization for BiGFuzzE approach. On the other hand, diagnoses and procedure codes are embedded in a vector space using Skip-gram model in BiGFuzzE+vector method. We observe that BiGFuzzE and BiGFuzzE+vector perform poorly when the value of λ is between 0.01 and 0.05. Both these methods demonstrate similar results in accuracy within the similar values of λ presented in Figure 10(a). Additionally, both methods demonstrate identical results when λ is set at 0.3 or more. Further, both the methods perform poorly for lower values of λ because highly dissimilar diagnoses codes are properly differentiated by both NMF and vector embeddings. As NMF factorizes the closeness matrix, which captures path lengths between diagnosis and procedure pairs, the diagnoses which are not commonly paired within a treatment share a longer path in the bipartite knowledge graph. Similarly, the diagnoses which are commonly paired with each other within a treatment share a shorter path in the bipartite knowledge graph. We observe similar phenomena when vector embeddings concept is used to represent diagnoses. The reason behind such phenomena is that both the most frequently and infrequently appearing codes are very well separated by the Skip-gram model applied on the inpatient claims. As our inpatient claim data has lower average number of codes, closeness matrix can properly capture the path lengths between a diagnosis and procedure pair. Another way to look at it is that claims with fewer diagnoses and procedure codes create a bipartite knowledge graph with fewer edges compared to claims with higher number of codes within claims. Therefore, we observe significant differences in accuracy and recall scores between BiGFuzzE and BiGFuzzE+vector methods for lower values of λ. BiGFuzzE demostrates better results compared to BiGFuzzE+vector within the values of λ between 0.05 and 0.03. The reason is that lower average number of codes create issues for Skip-gram model to associate different claims based on the neighborhood of codes within a predefined sliding window parameter. Overall, we observe insignificant number of false negatives in the results for both methods when λ is more than 0.3, which contributes to higher accuracy scores in Figure 10(a). We also notice insignificant false positives when the value of λ is 0.05 or more. As a result the precision scores in Figure 10(b) show exceptional results. However, the recall scores in Figure 10(c) follow accuracy score as it generates fewer false negatives when decision threshold $\lambda \leq 0.3$. It also shows insignificant false negatives when $\lambda > 0.3$.

Figure 11 presents the accuracy, precision, and recall scores for BiGFuzzE and BiGFuzzE+vector methods based on different decision threshold, λ, applied on outpatient claims data. We notice that BiGFuzzE+vector performs relatively better compared to BiGFuzzE based on accuracy and recall scores. As the average number of codes within a claim is higher in outpatient compared to inpatient claims, the Skip-gram model can differentiate between fraudulent and non-fraudulent claims. On the other hand, a diagnosis and procedure can share a path in the bipartite knowledge graph although they appeared in both fraudulent and non-fraudulent claims. Therefore, the results of BiGFuzzE show that in some cases, non-fraudulent claims are identified as fraudulent, resulting in false negatives. The false negatives affect accuracy and recall scores presented in Figures 11(a)



(a) Accuracy (b) Precision (c) Recall

Fig. 10. Parameter sensitivity with respect to λ on the inpatient dataset



(a) Accuracy (b) Precision (c) Recall

Fig. 11. Parameter sensitivity with respect to λ on the outpatient dataset

and 11(c), respectively. On the other hand, both `BiGFuzzE` and `BiGFuzzE+vector` demonstrate no false positive on outpatient claims data based on varying values of λ, resulting 100% precision scores in Figure 11(b).

VI. Conclusions

In this paper, we solve the problem of fraudulent healthcare claim identification using fuzzy bipartite graphs and matrix factorization techniques. We formulate the problem over claims with nominal information consisting of only diagnoses and procedure codes, because accessing richer datasets are often prohibited by law and present inconsistencies among different software systems. We employ the maximum reliability paths on fuzzy edges of the bipartite knowledge graph as a distance metric between diagnosis and procedure code pairs. In addition, our proposed approach adopts clinical code-level relation analysis within claims to perform fraud identification. We extend our proposed `BiGFuzzE` approach using vector embeddings of codes that substitutes matrix factorization subprocess and provides improved performance on claims with higher code frequencies. Our experimental results show that Bipartite Graph with Fuzzy Edges (`BiGFuzzE`) reaches an accuracy, precision, and recall scores of 95%, 100%, and 87%, respectively on the inpatient dataset acquired from CMS. Additionally, it demonstrates 97%, 100%, and 94% accuracy, precision, and recall scores, respectively on the outpatient dataset. We believe that the proposed problem formulation, medical code-level analysis, and solution will initiate new research on fraudulent claim identification using nominal, but definitive data in healthcare records.

References

[1] NHCAA, "The challenge of health care fraud," https://www.nhcaa.org/resources/health-care-anti-fraud-resources/the-challenge-of-health-care-fraud.aspx, 2020, accessed February, 2020.

[2] The Center for Medicare and Medicaid Services, "Nhe fact sheet," https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/NationalHealthExpendData/NHE-Fact-Sheet, 2020, accessed June, 2020.

[3] A. Wayne, "Health-care spending to reach 20% of us economy by 2021," *Bloomberg News, June 13*, 2012.

[4] National Health Care Anti-Fraud Association, "Consumer info and action," https://www.nhcaa.org/resources/health-care-anti-fraud-resources/consumer-info-action.aspx, 2020, accessed January, 2020.

[5] W. J. Rudman, J. S. Eberhardt, W. Pierce, and S. Hart-Hester, "Health-care fraud and abuse," *Perspectives in Health Information Management/AHIMA, American Health Information Management Association*, vol. 6, no. Fall, 2009.

[6] Find-A-Code, "Icd 10 codes, cpt codes, hcpcs codes, icd 9 codes - online encoder - medical billing and coding," https://www.findacode.com/index.html, 2020, accessed February, 2020.

[7] T. Bai, S. Zhang, B. L. Egleston, and S. Vucetic, "Interpretable representation learning for healthcare via capturing disease progression through time," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 43–51.

[8] Y. Wang and F. Tian, "Recurrent residual learning for sequence classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 938–943.

[9] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[10] W.-S. Yang and S.-Y. Hwang, "A process-mining framework for the detection of healthcare fraud and abuse," *Expert Systems with Applications*, vol. 31, no. 1, pp. 56–68, 2006.

[11] R. A. Bauder and T. M. Khoshgoftaar, "The detection of medicare fraud using machine learning methods with excluded provider labels," in *The Thirty-First International Flairs Conference*, 2018.

[12] R. Bauder, T. M. Khoshgoftaar, and N. Seliya, "A survey on the state of healthcare upcoding fraud analysis and detection," *Health Services and Outcomes Research Methodology*, vol. 17, no. 1, pp. 31–55, 2017.

[13] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using machine learning methods," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 858–865.

[14] R. A. Bauder, T. M. Khoshgoftaar, A. Richter, and M. Herland, "Predicting medical provider specialties to detect anomalous insurance claims," in *2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI)*. IEEE, 2016, pp. 784–790.

[15] R. A. Bauder and T. M. Khoshgoftaar, "Multivariate outlier detection in medicare claims payments applying probabilistic programming methods," *Health Services and Outcomes Research Methodology*, vol. 17, no. 3-4, pp. 256–289, 2017.

[16] ——, "A probabilistic programming approach for outlier detection in healthcare claims," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 347–354.

[17] I. D. Bross, "How to use ridit analysis," *Biometrics*, pp. 18–38, 1958.

[18] P. L. Brockett, R. A. Derrig, L. L. Golden, A. Levine, and M. Alpert, "Fraud classification using principal component analysis of ridits," *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 341–371, 2002.

[19] R. C. Mosley Jr and N. Kucera, "The use of analytics for claim fraud detection," in *SAS Global Forum*, 2014, p. 15.

[20] L. Settipalli and G. Gangadharan, "Wmtdbc: An unsupervised multivariate analysis model for fraud detection in health insurance claims," *Expert Systems with Applications*, vol. 215, p. 119259, 2023.

[21] H. Farbmacher, L. Löw, and M. Spindler, "An explainable attention network for fraud detection in claims management," *Journal of Econometrics*, vol. 228, no. 2, pp. 244–258, 2022.

[22] M. E. Haque and M. E. Tozal, "Identifying health insurance claim frauds using mixture of clinical concepts," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2356–2367, 2021.

[23] W. Zhang and X. He, "An anomaly detection method for medicare fraud detection," in *Big Knowledge (ICBK), 2017 IEEE International Conference on*. IEEE, 2017, pp. 309–314.

[24] I. Kose, M. Gokturk, and K. Kilic, "An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance," *Applied Soft Computing*, vol. 36, pp. 283–299, 2015.

[25] J. Wang and S. Luo, "Augmented beta rectangular regression models: A bayesian perspective," *Biometrical Journal*, vol. 58, no. 1, pp. 206–221, 2016.

[26] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software*, vol. 20, pp. 1–37, 2016.

[27] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of statistical software*, vol. 76, no. 1, 2017.

[28] R. A. Sowah, M. Kuuboore, A. Ofoli, S. Kwofie, L. Asiedu, K. M. Koumadi, and K. O. Apeadu, "Decision support system (dss) for fraud detection in health insurance claims using genetic support vector machines (gsvms)," *Journal of Engineering*, vol. 2019, 2019.

[29] T. K. Mackey, K. Miyachi, D. Fung, S. Qian, and J. Short, "Combating health care fraud and abuse: Conceptualization and prototyping study of a blockchain antifraud framework," *Journal of medical Internet research*, vol. 22, no. 9, p. e18623, 2020.

[30] A. Alnuaimi, A. Alshehhi, K. Salah, R. Jayaraman, I. A. Omar, and A. Battah, "Blockchain-based processing of health insurance claims for prescription drugs," *IEEE Access*, vol. 10, pp. 118 093–118 107, 2022.

[31] V. Chandola, S. R. Sukumar, and J. C. Schryver, "Knowledge discovery from massive healthcare claims data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1312–1320.

[32] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[33] J. Liu, E. Bier, A. Wilson, T. Honda, K. Sricharan, L. Gilpin, J. A. G. Gómez, and D. Davies, "Graph analysis for detecting fraud, waste, and abuse in healthcare data." in *AAAI*, 2015, pp. 3912–3919.

[34] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graphs," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[35] A. Rosenfeld, "Fuzzy graphs," in *Fuzzy sets and their applications to cognitive and decision processes*. Elsevier, 1975, pp. 77–95.

[36] R. T. Yeh and S. Bang, "Fuzzy relations, fuzzy graphs, and their applications to clustering analysis," in *Fuzzy sets and their applications to Cognitive and Decision Processes*. Elsevier, 1975, pp. 125–149.

[37] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[38] M. Gao, L. Chen, X. He, and A. Zhou, "Bine: Bipartite network embedding," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 715–724.

[39] M. Roosta, "Routing through a network with maximum reliability," *Journal of Mathematical Analysis and Applications*, vol. 88, no. 2, pp. 341–347, 1982.

[40] R. Bellman and R. Kalaba, "On k th best policies," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 582–588, 1960.

[41] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM (JACM)*, vol. 24, no. 1, pp. 1–13, 1977.

[42] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[43] M. Nijs, T. Smets, E. Waelkens, and B. De Moor, "A mathematical comparison of non-negative matrix factorization related methods with practical implications for the analysis of mass spectrometry imaging data," *Rapid Communications in Mass Spectrometry*, vol. 35, no. 21, p. e9181, 2021.

[44] M. E. Haque and M. E. Tozal, "Negative insurance claim generation using distance pooling on positive diagnosis-procedure bipartite graphs," *ACM Journal of Data and Information Quality (JDIQ)*, vol. 14, no. 3, pp. 1–26, 2022.

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[46] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

Md Enamul Haque is a Senior Research Data Scientist at the Spencer Center for Vision Research at Stanford University School of Medicine. He is also involved in eye care research at Office of Research and Development in U.S. Department of Veterans Affairs, Palo Alto, CA. He received his Ph.D. degree in Computer Science from the University of Louisiana at Lafayette. His general research area includes bioinformatics, data mining, NLP, and statistical learning to identify influential factors for eye disease among patients with a specific focus on research at the intersection of aging, dementia, and ocular disease. He is broadly interested in developing methods for precision medicine by applying graph machine learning and statistical modeling at the intersection of EHR, clinical trials, genomics, and related healthcare data.

Mehmet Engin Tozal is a Francis Patrick Clark/BORSF endowed Associate Professor in the School of Computing and Informatics at the University of Louisiana at Lafayette. He received his Ph.D. degree in Computer Science from the University of Texas at Dallas in 2012. Dr. Tozal's research focus has been on practical and theoretical aspects of complex systems and data. Specifically, he develops machine learning and graph theoretic approaches to study and solve various problems appearing in cybersecurity, computer networks, health informatics, decision support systems as well as data analysis and visualization domains.