Hcpcs2Vec: Healthcare Procedure Embeddings for Medicare Fraud Prediction

Justin M. Johnson and Taghi M. Khoshgoftaar
College of Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida 33431
jjohn273@fau.edu, khoshgof@fau.edu

2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC) | 978-1-7281-4146-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/CIC50333.2020.00026

***Abstract*—This study evaluates semantic healthcare procedure code embeddings on a Medicare fraud classification problem using publicly available big data. Traditionally, categorical Medicare features are one-hot encoded for the purpose of supervised learning. One-hot encoding thousands of unique procedure codes leads to high-dimensional vectors that increase model complexity and fail to capture the inherent relationships between codes. We address these shortcomings by representing procedure codes using low-rank continuous vectors that capture various dimensions of similarity. We leverage publicly available data from the Centers for Medicare and Medicaid Services, with more than 56 million claims records, and train Word2Vec models on sequences of co-occurring codes from the Healthcare Common Procedure Coding System (HCPCS). Continuous-bag-of-words and skip-gram embeddings are trained using a range of embedding and window sizes. The proposed embeddings are empirically evaluated on a Medicare fraud classification problem using the Extreme Gradient Boosting learner. Results are compared to both one-hot encodings and pre-trained embeddings from related works using the area under the receiver operating characteristic curve and geometric mean metrics. Statistical tests are used to show that the proposed embeddings significantly outperform one-hot encodings with 95% confidence. In addition to our empirical analysis, we briefly evaluate the quality of the learned embeddings by exploring nearest neighbors in vector space. To the best of our knowledge, this is the first study to train and evaluate HCPCS procedure embeddings on big Medicare data.**

***Keywords*—Semantic Embeddings, HCPCS, Word2Vec, Medicare, Fraud Detection, Big Data, XGBoost**

1. Introduction

The United States (US) Medicare program provides affordable health insurance to individuals 65 years and older, and other individuals with permanent disabilities [1]. There are currently more than 62.2 million U.S citizens enrolled in Medicare [2], and 2019 expenditures exceeded $796 billion [3]. The Federal Bureau of Investigation estimates that fraud accounts for up to 10% of all billings [4], i.e. up to $79 billion per year in the Medicare program. Some examples of fraud are billing for appointments that patients do not keep, billing for services not provided, or billing for services more complex than those performed. In an effort to reduce fraud, the Centers for Medicare and Medicaid Services (CMS) makes Medicare data sets publicly available for analysis [5]. In this study, we use the 2012–2017 Medicare Provider Utilization and Payment Data: Physician and Other Supplier Public Use File (Part B) data sets made available by CMS. We assign class labels to the Part B data using real-world fraud labels from the List of Excluded Individuals and Entities (LEIE) [6], and we train machine learning models to predict whether or not a provider is fraudulent.

Each record within the Part B claims data contains procedure-level statistics for a specific provider over a given year, including the number of times the procedure was performed, the number of beneficiaries receiving the service, and the average amount billed to Medicare. Providers are identified by their National Provider Identification number (NPI) and the procedures performed are identified by standardized codes defined by the Healthcare Common Procedure Coding System (HCPCS) [7]. The Part B data used in this study includes 7,527 unique HCPCS procedure codes. Table 1 lists four examples of HCPCS procedure codes and their descriptions. Traditionally, related works have used one-hot vectors to encode the HCPCS attribute when training machine learning algorithms to classify fraudulent providers. This produces very large and sparse feature vectors that increase model complexity and pose challenges related to the curse of dimensionality [8]. These sparse one-hot vectors also fail to capture any form of similarity between procedure codes, i.e. codes are either equal or equidistant from each other. We address this by converting the categorical HCPCS procedure codes to low-rank continuous vectors that encode the semantic relationships that exist between similar procedures.

TABLE 1. HCPCS PROCEDURE CODE EXAMPLES

Code	Description
70551	MRI scan brain
90791	Psychiatric diagnostic evaluation
G0008	Administration of influenza virus vaccine
J0696	Injection, ceftriaxone sodium, per 250 mg

The concept of using distributed representations of HCPCS procedure codes is inspired by the success of word

978-1-7281-4146-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CIC50333.2020.00026

embeddings in natural language processing. Mikolov et al. [9] proposed two Word2Vec models for efficiently learning high-quality representations of words from sequences co-occurring words. Pennington et al. [10] later proposed Global Vectors for Word Representation (GloVe), a method for generating word embeddings from a global word-word co-occurrence matrix. From these, a number of more advanced embedding techniques have been developed and have transformed how complex natural language processing problems are solved [11], [12]. In this study, we convert the Part B Medicare data to a corpus of co-occurring procedure code sequences and use Word2Vec models to learn meaningful representations for procedure codes.

Three sets of semantic HCPCS embeddings are evaluated on the Medicare fraud classification problem using Extreme Gradient Boosting (XGBoost) [13] classifiers. First, we use the continuous-bag-of-words (CBOW) and skip-gram (SG) models presented by Mikolov et al. to learn two sets of procedure embeddings over a range of embedding and window sizes. Next, we leverage pre-trained procedure code embeddings that have been made publicly available in related works [14]. Finally, we compare the performance of these three sets of embeddings to three baseline representations using the area under the receiver operating characteristic curve (AUC) [15] and geometric mean (G-Mean) [16]. One baseline model excludes the HCPCS attribute entirely (NONE), another uses traditional one-hot vectors of length 7,527 (ONEHOT), and the third compresses the one-hot vectors to 75-dimensional random vectors drawn from a uniform distribution (RAND). Six runs of five-fold cross-validation are used to perform statistical tests. Tukey's Honestly Significant Difference (HSD) test and confidence intervals are used to show that semantic code embeddings significantly outperform all three baseline methods on Medicare fraud classification with significance level $\alpha = 0.05$. A nearest-neighbors analysis of CBOW and SG embeddings further show that the learned distributed representations capture meaningful semantics.

Our primary contribution is the application of Word2Vec models to learn semantic HCPCS procedure code embeddings. We provide step-by-step instructions demonstrating how to construct these embeddings from publicly available Medicare data, and we evaluate these embeddings against traditional embedding techniques on a fraud classification problem. Given the success of our results, future works can continue down this path and evaluate the use of more advanced embedding techniques, e.g. neural and graph-based attention models [17]. To the best of our knowledge, this is the first study to train and evaluate multiple sets of HCPCS embeddings using publicly available Medicare data.

The remainder of this paper is structured as follows. In Section 2, we review previous work related to medical concept embeddings and Medicare fraud detection. Section 3 describes the data set used in this study, the three embedding techniques employed, and the experiment design used for evaluation. We discuss our results in Section 4 and conclude with areas for future work in Section 5.

2. Related Work

2.1. Medicare Fraud Prediction

The Medicare fraud detection task has received a lot of attention in recent years. Bauder and Khoshgoftaar [18] estimate Medicare payments using five regression models and flag fraudulent providers by comparing actual payments to estimated payments. Ko et al. [19] use a linear regression model to analyze the variability of service utilization and payments. Branting et al. [20] derive graphs from Medicare Part B and Part D claims data and then extract features from these graphs for fraud classification. Each provider, prescription drug, and HCPCS procedure code are represented as graph nodes and they are linked by the relations in historical Medicare claims data. Behavioral similarity and geospatial colocation features are extracted from the graph and a decision tree model is used to classify fraudulent providers. In [21], Herland et al. use a Naive Bayes learner to predict Medicare provider specialties from HCPCS procedure occurrences. Leveraging fraud labels from the LEIE data set, providers assigned to the wrong specialty type are classified as fraudulent. Chandola et al. [22] model providers as documents by constructing provider-diagnosis matrices from claims data, and then use Latent Dirichlet Allocation to identify 20 hidden topics. The authors then show how specific topics extracted from the claims data capture a vast majority of the fraudulent providers. Herland et al. [23] explore fraud prediction using three 2012–2015 CMS Medicare data sets, i.e. Part B, Part D, and DMEPOS. Fraudulent providers are identified using the LEIE data set and the feature set includes provider activity statistics, e.g. the average amount billed, average amount paid, and average number of beneficiaries treated. Hancock and Khoshgoftaar [24] compare the Extreme Gradient Boosting (XGBoost) classifier to the CatBoost classifier [25] using the Part B data set. We extended work by Herland et al. and improved classification performance by replacing one-hot provider type attributes with learned dense emebeddings [26]. In another study [27], we evaluate deep neural networks and various techniques for addressing class imbalance using the Part B data set. Data-level and algorithm-level methods are used to treat class imbalance, and results show that balancing training data with random over-sampling (ROS) and random under-sampling (RUS) maximizes performance with an average ROC AUC of 0.8506. Of these related works, most do not unitilize the HCPCS procedure code attribute in their model and none of the works mentioned thus far employ procedure code embeddings.

Fursov et al. [28] propose an embedding technique for identifying fraudulent healthcare claims by treating the claims data as a corpus. Individual treatments within a claim are mapped to a vector representation, and the sequences of treatments within a claim are then aggregated to create a final embedding space. The proposed embedding technique is shown to outperform bag-of-words and term frequency-inverse document frequency methods. It is not clear if the ground truth labels used by Fursov et al. are representative

of fraud, however, as the claims are labeled as fraudulent if the final bill required a correction. Furthermore, the vocabulary of treatments is significantly smaller than the HCPCS vocabulary used in our study, i.e. 2,205 treatments vs. 7,527 HCPCS codes. Therefore, we expand on this work by using known fraudulent providers, comparing multiple embedding techniques to multiple baseline representations, and by exploring a range of embedding and window sizes. In addition, we provide step-by-step instructions on how to generate HCPCS embeddings from publicly available big Medicare data so that others may use them in subsequent works.

2.2. Medical Concept Embeddings

Several research groups have extended word embedding techniques to the biomedical domain and learned new representations for medical concepts, e.g. disease codes, medications, procedures, and laboratory tests. De Vine et al. [29] produce medical concept embeddings from free text in clinical records and medical journal abstracts. Sequences of medical concepts are first created by mapping the free text to concepts defined in the UMLS Metathesaurus using MetaMap [30]. SG models are trained on the sequences of medical concepts using a range of embedding and window sizes. Semantic similarity results are evaluated against six baseline methods using two data sets of similarity pairs that were produced by expert medical judges. The SG embeddings perform best overall according to the Pearson correlation measures and on average larger embedding and window sizes improve performance. Choi et al. [14] use SG models to train medical concept embeddings from multiple sources, and results are compared to the medical concept embeddings from medical journals (MCEMJ) provided by De Vine et al. [29]. Medical concept embeddings from medical claims (MCEMC) are trained on claims data that spans over four million patients between 2005 and 2013. Medical concept embeddings from clinical narratives (MCECN) are trained on concept co-occurrence matrices from 20 million clinical notes across 19 years of data. The MCEMJ embeddings score the highest overall on a medical concept similarity measure and the MCEMC embeddings score the highest overall on a medical relatedness measure. Beam et al. [31] present cui2vec embeddings, the largest set of medical concept embeddings known to date. Cui2vec embeddings are trained using multi-modal data, i.e. claims data from 60 million patients, 20 million clinical notes, and 1.7 million medical journal articles. The authors create embeddings from the combined sources of data using the GloVe, Word2Vec, and PCA algorithms. Embeddings are compared using five benchmarks, and the Word2Vec embeddings are shown to perform best overall with an embedding size of 500. Each of these studies use semantic similarity measures as benchmarks to evaluate their embeddings, whereas we focus specifically on the downstream task of classifying Medicare fraud.

Several works supplement their medical concept embeddings with ontologies, or relational knowledge graphs, e.g. the Clinical Classifications Software (CSS) categorization scheme [32]. Choi et al. [17] proposed the GRaph-based Attention Model (GRAM), which learns representations for diagnosis codes from a combination of CSS ontological ancestors via a neural attention weighting mechanism. GRAM is compared to five baseline methods using two next-visit diagnosis prediction tasks and a heart failure prediction task, and results suggest GRAM outperforms baseline methods on low-frequency diseases and small data sets. Ma et al. [33] propose the Knowledge-Based Atttention Model (KAME) for predicting patients' future health conditions. Unlike GRAM, KAME exploits medical knowledge throughout the whole prediction process, i.e. code representations, visit embeddings, and down-stream predictions. Ma et al. show that KAME outperforms GRAM and three other baseline methods on three diagnosis prediction tasks. Song et al. [34] extend the GRAM model with Medical Concept Embeddings with Multiple Ontological REpresentations (MMORE). MMORE combats the inconsistencies between EHR data and medical ontologies by allowing multiple representations of hierarchical ontology concepts to be learned and outperforms GRAM. In each of these works, ontology relationships help align the learned embeddings with expert medical knowledge and improve the quality of low-frequency concept embeddings.

Many of these related works focus on learning embeddings for diagnosis codes and medications for the purpose of evaluating semantic similarity and predicting future diagnosis. In our study, we focus specifically on using low-dimensional representations for HCPCS procedure codes to improve Medicare fraud classification results. Furthermore, we are the first to provide step-by-step instructions for training distributed representations of HCPCS procedure codes from big Medicare data. We include the MCEMC embeddings provided by Choi et al. [14] in our evaluation because they are readily available for encoding HCPCS procedure codes and they do not require mapping HCPCS codes to CUI codes. To the best of our knowledge, this is the first study to train and evaluate multiple HCPCS procedure code embeddings on a Medicare fraud classification task.

3. Methods

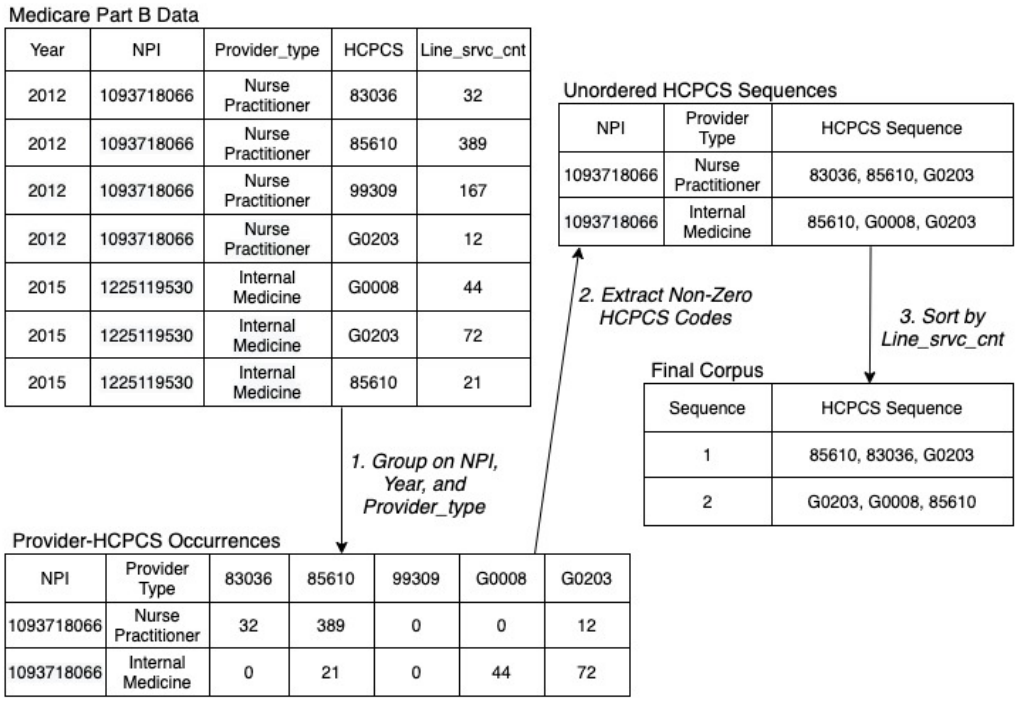
3.1. Medicare Part B Data

This study uses the 2012–2017 Medicare Part B data that is publicly available on the CMS website in comma delimited format [5]. The Part B data set contains more than 56 million records, and it includes a combination of provider-level and procedure-level features. A description of each feature is provided in Table 2. There exists one row of data for each NPI, Provider_type, HCPCS, and Year combination. The provider type attribute consists of 123 values that describe a provider's specialty, e.g. Internal Medicine, Nurse Practitioner, and Urology. The HCPCS code attribute includes 7,527 procedure codes that identify specific procedures performed by a provider. The numeric attributes describe the provider's billing activity relative to a HCPCS

TABLE 2. DESCRIPTION OF PART B FEATURES

Feature	Description	Type
NPI	Unique provider identification number	Categorical
HCPCS	Procedure/service code	Categorical
Provider_type	Medical provider's specialty (or practice)	Categorical
Nppes_provider_gender	Provider's gender	Categorical
Year	Year of billing activity	Categorical
Line_srvc_cnt	Number of procedures/services the provider performed	Numeric
Bene_unique_cnt	Number of distinct beneficiaries receiving the service	Numeric
Bene_day_srvc_cnt	Number of distinct beneficiary/per day services	Numeric
Average_submitted_chrg_amt	Average of charges that provider submitted for the HCPCS	Numeric
Average_medicare_payment_amt	Average payment made to a provider per claim for the HCPCS	Numeric
Exclusion	Fraud labels from the LEIE data set	Categorical

Figure 1. Creating HCPCS Corpus from CMS Part B Data



code for a given year. The Exclusion label is the class label, i.e. fraudulent vs non-fraudulent, and it is mapped to the Part B data by joining the LEIE data set on the provider NPI. The LEIE data set is a list of providers that have been excluded from practice due to fraudulent behaviors, as determined by the Office of Inspector General [6]. Specific instructions for assigning class labels and preprocessing features can be found in earlier work by Herland et al. [21].

3.2. HCPCS Embedding Techniques

Word2Vec algorithms [9] are used to learn distributed representations for HCPCS procedure codes from Medicare Part B data. The Word2Vec algorithms follow the distributional hypothesis, which states that the degree of semantic similarity between two concepts can be modeled as a function of the degree of overlapping context [35]. Given sequences of co-occurring words, Word2Vec learns to represent each word w as a d-dimensional vector $\vec{w}$, such that words that are similar to each other have similar vector representations. The CBOW implementation of Word2Vec aggregates the distributed representations of context terms, or neighboring terms, and learns to predict the center target term from the aggregated context. The SG implementation uses the distributed representation of the input term to estimate the probability distribution of neighboring context terms. The context window radius L is a hyperparameter that defines the size of the context to use during training, and the embedding dimension size d defines the size of the learned representations. When the Word2Vec model's objective function converges, word embeddings are extracted from the model's hidden layer and used freely on a variety of downstream data mining and machine learning tasks.

Before we can apply Word2Vec, we must first create a sufficiently large corpus of sequences comprised of co-occurring HCPCS codes. We use more than 56 million

records from the 2012–2017 Medicare data to create this corpus of HCPCS sequences. First, we remove rows that are missing either HCPCS or Line_srvc_cnt attributes. Next, we create a provider-hcpcs occurrence matrix by grouping on the NPI, Year, and Provider_type attributes. This yields a list of HCPCS procedures performed by a given provider over a given year, and the total number of times each procedure was performed that year. Finally, we sort each HCPCS procedure sequence by the Line_srvc_cnt attribute, reordering the sequence so that procedures that occur with similar frequencies are neighbors. This ordering explicitly modifies the contexts used by Word2Vec and allows the learned embeddings to encode one or more dimensions related to the frequency with which a HCPCS procedure occurs. Therefore, the resulting HCPCS sequences produce context windows representative of specific providers, provider types, and frequencies. The final corpus of HCPCS co-occurrences contains 4.7 million sequences of HCPCS procedure codes. This process is outlined visually in Figure 1.

The CBOW and SG models are trained on the corpus of 4.7 million HCPCS sequences using the Gensim Python package v3.8.0 [36]. The SG model uses $min\_count = 2$, $iters = 100$, and negative sampling with $negative = 5$. The CBOW model uses $min\_count = 2$, $iters = 200$, and aggregates context representations by taking the mean. These settings performed best during preliminary experiments and validation. We use window radius sizes $L \in \{5, 10\}$ and embedding sizes $d \in \{75, 150, 300\}$ for both embedding techniques.

We also evaluate MCEMC pre-trained procedure code embeddings from Choi et al. [14]. The MCEMC embeddings were trained on claims data from four million patients between 2005–2013. Choi et al. partition the temporal claims data into intervals, remove duplicate terms, and use the SG model to produce embeddings of size 300. The MCEMC embedding set contains procedure code embeddings for 6,367 out of the 7,527 unique HCPCS codes in the Medicare Part B data set, i.e. 84.6% coverage. We consider two options for encoding out-of-vocab, or unknown (UNK), procedure codes in MCEMC. The first option (MCEMC) treats all out-of-vocab codes as as the same term, "unknown"" or UNK, and encodes them using a vector of zeros. The second approach (MCEMC_RAND) encodes each unique UNK code with a unique 300-dimensional random vector drawn from the uniform distribution.

3.3. Fraud Classification

The CBOW, SG, and MCEMC procedure code embeddings are evaluated on the Part B fraud classification task. The labelled fraud data set is constructed by mapping real-world fraud labels from the public LEIE data set [6] using provider NPI numbers [23]. After adding class labels, we remove the NPI and year attributes from the original feature set listed in Table 2. The Provider_type and Nppes_provider_gender attributes are one hot encoded, and the HCPCS code is replaced by each respective embedding technique. The HCPCS encoding technique significantly affects the total number of features used to model fraudulent activity. For example, when HCPCS codes are one-hot encoded there are a total of 7633 features, but when a HCPCS embedding with $d = 75$ is used only 181 features are required.

From the 56 million data points in the 2012–2017 Part B data set, only 36 thousand are labelled as fraudulent, i.e. just 0.06%. To address the challenges related to highly imbalanced big data [37], we employ a simple data sampling technique to reduce the size of the majority class [38]. We combine all 36 thousand positive samples with a random sample (without replacement) from the non-fraudulent class to create sample sets comprised of 4 million records. We select 4 million as the sample size because preliminary results reveal diminishing returns with additional data. As shown in Table 3, this under-sampling increases the size of the positive class to 0.91%. This minority class size is approximately equal to the minority class size of the best performing under-sampling results in a related Medicare fraud classification study [39], [40]. We do not treat the class imbalance problem further because it is outside the scope of this work.

We evaluate HCPCS embedding techniques on the Medicare fraud classification problem using the Extreme Gradient Boosting (XGBoost) classifier. Chen and Guestrin [13] describe XGBoost as a scalable tree boosting system that is used widely in machine learning challenges and is well known for achieving state-of-the-art performance on a variety of tasks. XGBoost models are trained with the xgboost v.0.90 Python package using a $max\_depth = 8$ and default values for all remaining hyperparameters, e.g. $n\_estimators = 100$ and $learning\_rate = 0.1$. Six runs of five-fold cross-validation is used for each experiment. A new set of 4 million instances is sampled for each iteration of five-fold cross-validation to account for any deviations caused by subsampling the non-fraudulent class. All experiments are conducted using a high-performance computing environment running Scientific Linux 7.4 (Nitrogen) [41]. Jobs are dispatched onto CPU nodes with 20 Intel(R) Xeon(R) CPU E5-2660 v3 2.60GHz processors and 128GB of RAM.

We visualize the average AUC scores for each embedding technique using boxplots and we estimate the statistical significance of our results using Tukey's HSD test with $\alpha = 0.05$. Tukey's HSD test is a multiple comparison procedure that determines which method means are statistically different from each other by identifying differences that are greater than the expected standard error [42]. Embedding techniques are assigned to alphabetic groups based on the statistical difference of AUC means, e.g. group a is significantly different from group b. We also report the AUC and G-Mean 95% confidence intervals for the three top performing embedding techniques. Similar to related works [43], [44], we use a classification threshold equal to the positive class prior (0.0091 = 0.91%) when assigning class labels from posterior probabilities.

Figure 2. AUC Results and HSD Groups

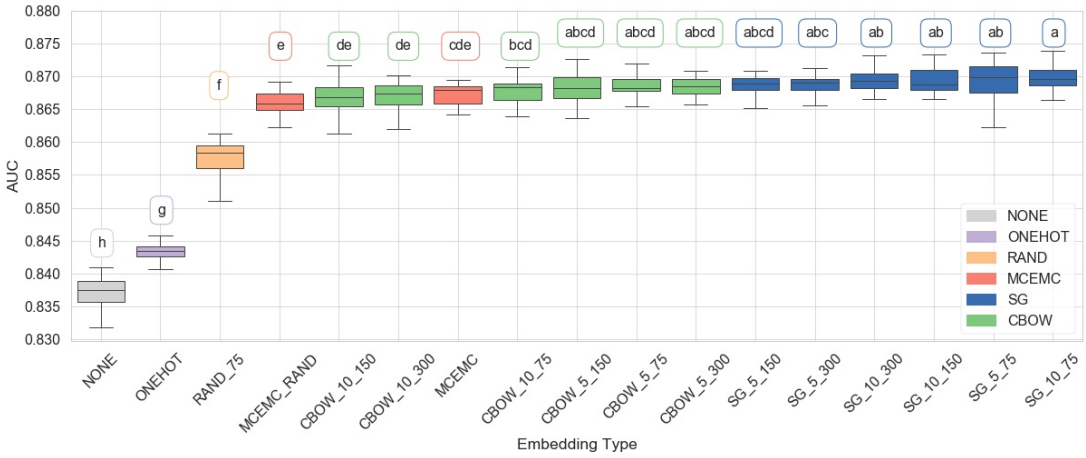


TABLE 3. PART B DATA SET SIZES

Data Set	Total Samples	Fraudulent Samples	% Fraudulent
Original Data	56,818,165	36,548	0.06%
Sampled Data	4,000,000	36,548	0.91%

4. Results and Discussion

First, we evaluate HCPCS embedding techniques on the Medicare Part B fraud classification task. Figure 2 illustrates the AUC results taken over six runs of five-fold cross-validation, i.e. 30 runs per method. The window and embedding sizes of each method are denoted using underscores, e.g. SG_5_300 corresponds to a skip-gram model with $L = 5$ and $d = 300$. Tukey's HSD groups are included to identify methods whose means are significantly different.

Removing the HCPCS attribute entirely (NONE) performs the worst overall with a mean AUC score of 0.837. Traditional one-hot encoding of HCPCS codes (ONEHOT) increases the mean AUC score to 0.843, and random 75-dimensional embeddings drawn from the uniform distribution $[0, 1]$ (RAND) increase performance to 0.857. All three of these baseline results belong to different HSD groups and have significantly different means. We can conclude from these results that the HCPCS attribute has a significant affect on performance and compressing the original one-hot vectors ($d = 7527$) down to 75 dimensions improves performance.

All three sets of semantic embeddings further increase the average AUC and significantly outperform NONE, ONEHOT, and RAND. From Figure 2, we observe that SG consistently outperforms CBOW across all window and embedding sizes. Results also show that many of the semantic embedding techniques contain overlapping HSD groups. For example, 9 of 14 semantic embedding techniques belong to HSD group *a*. MCEMC and MCEMC_RAND embedding results belong to HSD groups *cde* and *e*, respectively, and MCEMC overlaps with all CBOW results. The top four SG embeddings do not belong to group *c* and significantly outperform MCEMC and several CBOW methods. Finally, there is no obvious affect caused by the window and embedding size parameters of SG and CBOW. The SG embeddings perform best overall and all semantic embedding techniques improve upon ONEHOT and RAND with mean AUC scores in the range 0.867–0.870.

Table 5 lists the 95% confidence intervals for the AUC and G-mean scores of the best performing embeddings taken from each category. Again, we observe that SG_10_75 performs best overall with AUC and G-Mean intervals of (0.869, 0.870) and (0.781, 0.783), respectively. While MCEMC and CBOW_5_300 perform approximately the same as SG, their difference in AUC means is significant as their confidence intervals do not overlap. With an increase in G-Mean scores from 0.754 to 0.783, we conclude that distributed representations of HCPCS procedure codes perform significantly better than one-hot vector representations.

Next, we evaluate the HCPCS procedure code embeddings that were trained on Medicare Part B data by inspecting the nearest neighbors using the cosine similarity measure [45]. Figure 4 lists four random HCPCS codes and the nearest neighbor in embedding space for the CBOW_5_300 and SG_10_75 representations, i.e. the two best performing embeddings. As hoped, we see that each example procedure embedding has a nearest neighbor that is very similar. For example, the code 23465 used to document repair of should joint procedures has nearest neighbors that are both related to should procedures. In another example, code 71030 identifies chest x-ray procedures. We see that the SG nearest neighbor is more closely related than the CBOW neighbor, as it also pertains to chest x-rays. Overall, these results show that both the CBOW and SG embedding techniques have produced distributed representations that capture some concept of similarity.

TABLE 4. HCPCS EMBEDDINGS AND NEAREST NEIGHBORS

Embedding	HCPCS Code	Nearest Neighbor (Cosine Similarity)
CBOW	20926 - Tissue graft	20900 - Small bone graft from any donor area
SG		15770 - Creation of skin, fat and muscle graft
CBOW	23465 - Repair of	23929 - Shoulder procedure
SG	shoulder joint	24301 - Relocation of muscle or tendon of upper arm or elbow
CBOW	84132 - Blood	82465 - Cholesterol level
SG	potassium level	85014 - Red blood cell concentration measurement
CBOW	71030 - X-ray of chest,	70030 - X-ray of eye
SG	minimum of 4 views	71101 - X-ray of ribs with chest minimum of 3 views

TABLE 5. AUC AND G-MEAN 95% CONFIDENCE INTERVALS

Embedding	L	d	AUC	G-Mean
ONEHOT	–	7527	(0.843, 0.844)	(0.754, 0.756)
MCEMC	5	300	(0.867, 0.868)	(0.778, 0.780)
CBOW	5	300	(0.868, 0.869)	(0.780, 0.783)
SG	10	75	(0.869, 0.870)	(0.781, 0.783)

5. Conclusion

In this study, we evaluate and train HCPCS procedure code embeddings using big Medicare Part B data that has been made publicly available by the CMS. We create sequences of HCPCS procedure codes from more than 56 million Medicare data points, and we learn distributed representations for the procedure codes using Word2Vec algorithms. Embeddings are created using the CBOW and SG implementations of Word2Vec using a range of window and embedding sizes. The embeddings are empirically evaluated on the Medicare fraud classification task and results are compared to traditional one-hot representations and publicly available pre-trained embeddings (MCEMC). Six runs of five-fold cross-validation and statistical analysis are used to compare the AUC and G-Mean results of each embedding using the XGBoost classifier. We found that all three semantic embeddings perform significantly better than traditional one-hot representations, and that the SG embeddings perform best overall. We also saw significant improvements over one-hot vectors when using random embedding vectors that were drawn from the uniform distribution $[0, 1]$, suggesting that the high cardinality of HCPCS procedure codes degrade XGBoost performance. Finally, we explore the nearest neighbors of several HCPCS procedure codes and show that similar procedures are represented by similar vectors.

In future works, we intend to apply the HCPCS procedure code embeddings to other downstream tasks, e.g. diagnosis prediction, readmission prediction, and semantic search. In addition, these embeddings can be evaluated across a wider range of machine learning algorithms, and results can be compared to models better designed for high-dimensional categorical variables. There is also an opportunity to explore additional embedding techniques like GloVe and character-level embeddings.

Acknowledgments

The authors would like to thank the reviewers in the Data Mining and Machine Learning Laboratory at Florida Atlantic University.

References

[1] U.S. Government, U.S. Centers for Medicare & Medicaid Services. The official u.s. government site for medicare. [Online]. Available: https://www.medicare.gov/

[2] Centers for Medicare & Medicaid Services. (2019) Medicare enrollment dashboard. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Dashboard/Medicare-Enrollment/Enrollment%20Dashboard.html

[3] Centers For Medicare & Medicaid Services. (2020) Trustees report & trust funds. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/ReportsTrustFunds/index.html

[4] L. Morris, "Combating fraud in health care: An essential component of any cost containment strategy," *Health affairs (Project Hope)*, vol. 28, pp. 1351–6, 09 2009.

[5] Centers For Medicare & Medicaid Services. (2019) Medicare provider utilization and payment data. [Online]. Available: https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/medicare-provider-charge-data

[6] Office of Inspector General. (2019) Leie downloadable databases. [Online]. Available: https://oig.hhs.gov/exclusions/exclusions_list.asp

[7] Centers For Medicare & Medicaid Services. (2018) Hcpcs general information. [Online]. Available: https://www.cms.gov/Medicare/Coding/MedHCPCSGenInfo/index.html

[8] L. Chen, *Curse of Dimensionality*. Boston, MA: Springer US, 2009, pp. 545–546. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_133

[9] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[10] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[11] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *ArXiv*, vol. abs/1802.05365, 2018.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[13] T. Chen and C. Guestrin, "Xgboost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[14] Y. Choi, C. Y.-I. Chiu, and D. A. Sontag, "Learning low-dimensional representations of medical concepts," *AMIA Summits on Translational Science Proceedings*, vol. 2016, pp. 41 – 50, 2016.

[15] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, vol. 43-48, 12 1999.

[16] R. Jain, *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.*, ser. Wiley professional computing. Wiley, 1991.

[17] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: Graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 787–795. [Online]. Available: https://doi.org/10.1145/3097983.3098126

[18] R. A. Bauder and T. M. Khoshgoftaar, "A novel method for fraudulent medicare claims detection from expected payment deviations (application paper)," in *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, July 2016, pp. 11–19.

[19] J. Ko, H. Chalfin, B. Trock, Z. Feng, E. Humphreys, S.-W. Park, B. Carter, K. D Frick, and M. Han, "Variability in medicare utilization and payment among urologists," *Urology*, vol. 85, 03 2015.

[20] L. K. Branting, F. Reeder, J. Gold, and T. Champney, "Graph analytics for healthcare fraud risk estimation," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug 2016, pp. 845–851.

[21] M. Herland, R. A. Bauder, and T. M. Khoshgoftaar, "Medical provider specialty predictions for the detection of anomalous medicare insurance claims," in *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, Aug 2017, pp. 579–588.

[22] V. Chandola, S. R. Sukumar, and J. C. Schryver, "Knowledge discovery from massive healthcare claims data," in *KDD*, 2013.

[23] M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, "Big data fraud detection using multiple medicare data sources," *Journal of Big Data*, vol. 5, no. 1, p. 29, Sep 2018. [Online]. Available: https://doi.org/10.1186/s40537-018-0138-3

[24] J. Hancock and T. M. Khoshgoftaar, "Medicare fraud detection using catboost," in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, 2020, pp. 97–103.

[25] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 6639–6649.

[26] J. Johnson and T. M. Khoshgoftaar, "Semantic embeddings for medical providers and fraud detection," *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 224–230, 2020.

[27] J. M. Johnson and T. M. Khoshgoftaar, "Medicare fraud detection using neural networks," *Journal of Big Data*, vol. 6, no. 1, p. 63, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0225-0

[28] I. Fursov, A. Zaytsev, R. Khasyanov, M. Spindler, and E. Burnaev, "Sequence embeddings help to identify fraudulent cases in healthcare insurance," *ArXiv*, vol. abs/1910.03072, 2019.

[29] L. De Vine, G. Zuccon, B. Koopman, L. Sitbon, and P. Bruza, "Medical semantic similarity with a neural language model," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1819–1822. [Online]. Available: https://doi.org/10.1145/2661829.2661974

[30] A. Aronson and F.-M. Lang, "An overview of metamap: Historical perspective and recent advances," *Journal of the American Medical Informatics Association : JAMIA*, vol. 17, pp. 229–36, 05 2010.

[31] A. L. Beam, B. Kompa, I. Fried, N. P. Palmer, X. Shi, T. Cai, and I. S. Kohane, "Clinical concept embeddings learned from massive sources of medical data," *ArXiv*, vol. abs/1804.01486, 2018.

[32] H. C. H. Cost and U. P. (HCUP), "Clinical classifications software (ccs) for icd-9-cm," www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp, 2017. [Online]. Available: www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

[33] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "Kame: Knowledge-based attention model for diagnosis prediction in healthcare," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 743–752. [Online]. Available: https://doi.org/10.1145/3269206.3271701

[34] L. Song, C. W. Cheong, K. Yin, W. K. Cheung, B. C. M. Fung, and J. Poon, "Medical concept embedding with multiple ontological representations," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 4613–4619. [Online]. Available: https://doi.org/10.24963/ijcai.2019/641

[35] Z. S. Harris, "Distributional structure," *¡i¿WORD¡/i¿*, vol. 10, no. 2-3, pp. 146–162, 1954. [Online]. Available: https://doi.org/10.1080/00437956.1954.11659520

[36] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[37] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *Journal of Big Data*, vol. 5, no. 1, p. 42, 2018. [Online]. Available: https://doi.org/10.1186/s40537-018-0151-6

[38] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 935–942. [Online]. Available: https://doi.org/10.1145/1273496.1273614

[39] J. M. Johnson and T. M. Khoshgoftaar, "Deep learning and data sampling with imbalanced big data," in *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, 2019, pp. 175–183.

[40] J. M. Johnson and T. M. Khoshgoftaar, "The effects of data sampling with deep learning and highly imbalanced big data," *Information Systems Frontiers*, 2020. [Online]. Available: https://doi.org/10.1007/s10796-020-10022-7

[41] S. Linux. (2014) About. [Online]. Available: https://www.scientificlinux.org/about/

[42] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, vol. 5, no. 2, pp. 99–114, 1949. [Online]. Available: http://www.jstor.org/stable/3001913

[43] J. M. Johnson and T. M. Khoshgoftaar, "Deep learning and thresholding with class-imbalanced big data," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 755–762.

[44] J. M. Johnson and T. M. Khoshgoftaar, "Thresholding strategies for deep learning with highly imbalanced big data," in *Deep Learning Applications, Volume 2*, A. Wani, T. M. Khoshgoftaar, and V. Palade, Eds. Springer, Singapore, 2020, pp. 199–227.

[45] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008. [Online]. Available: http://nlp.stanford.edu/IR-book/information-retrieval-book.html