2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI) | 978-1-6654-3875-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/IRI51335.2021.00055

Encoding Techniques for High-Cardinality Features and Ensemble Learners

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

***Abstract*—This study evaluates the classification performance of five encoding techniques for high-cardinality categorical features. Encoding techniques are tested using popular bagging and boosting ensemble methods on the latest Medicare Part B fraud classification data set, where the healthcare procedure code feature includes 7,752 unique values. To provide an additional baseline, we also evaluate these encodings on the multilayer perceptron. One-hot encodings are compared to a baseline aggregated encoding that excludes the procedure code feature to determine if the categorical feature significantly affects performance. Next, LightGBM and CatBoost's built-in strategies for categorical feature handling are compared to Hcpcs2Vec embeddings, distributed representations of procedures that encode semantic similarities. Statistical tests show that the inclusion of the categorical feature significantly improves performance for all ensemble learners when a one-hot representation is not used. Results also show that the XGBoost learner with Hcpcs2Vec encodings perform best overall with an average AUC of 0.8715. Our comparison of diverse encoding techniques for the high-dimensional categorical feature makes this study a unique contribution in the areas of ensemble learning and healthcare fraud prediction.**

***Keywords*—Categorical Features, High Cardinality, Ensemble Learners, Encoding, Semantic Embeddings, Medicare, Big Data, Fraud Detection**

1. Introduction

The United States (U.S.) Medicare program offers affordable health insurance for the elderly population and certain 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]. It is estimated that fraud accounts for up to 10% of all billings [4], i.e. up to $79 billion per year in the Medicare program. Examples of fraud include billing for services not provided, for appointments that patients did not keep, or for services more complex than those performed. To increase transparency and reduce fraud, the Centers for Medicare and Medicaid Services (CMS) makes yearly Medicare data sets publicly available for analysis [5]. In this study, we use the 2012–2018 Medicare Provider Utilization and Payment Data: Physician and Other Supplier Public Use File (Part B) data sets made available by CMS. To the best of our knowledge, this is the first study to expand Medicare fraud experiments to include the 2018 data set. We map fraudulent and non-fraudulent 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 ensemble learners to predict whether or not a provider is fraudulent.

Medicare records contain a variety of provider-level and procedure-level features, including a procedure code that is standardized by the Healthcare Common Procedure Coding System (HCPCS). The 2012–2018 Part B set includes 7,752 unique HCPCS procedure codes in total. Table 1 lists four examples of HCPCS procedure codes and their descriptions. Traditionally, related works have excluded the HCPCS attribute when training machine learning algorithms to detect fraudulent providers. We suspect this is because of its high cardinality, where a one-hot encoding would add 7,752 new features to an already big data set containing more than 66 million records. Instead, most previous works have used an aggregation method to group all of a provider's procedure-level statistics into a single row using summary statistics. We strongly believe that this aggregation incurs information loss, and that the inclusion of the HCPCS attribute will improve the fraud detection rates.

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

Five different encoding techniques are used to incorporate the high-cardinality HCPCS feature in this study. The Aggregate method from previous works [7] is used to serve as our baseline. The one-hot method uses a one-hot encoding to represent the HCPCS feature using a sparse representation that helps reduce memory requirements. The CatBoost and LightGBM encodings use each learner's respective automatic categorical feature handling capabilities, including ordered target statistics (TS) and exclusive feature bundling (EFB). Finally, we employ semantic embeddings using Hcpcs2Vec, distributed representations for procedure

978-1-6654-3875-9/21/$31.00 ©2021 IEEE
DOI 10.1109/IRI51335.2021.00055

codes that we have proposed in previous works [8]. We describe each of these encoding techniques in detail in Section 3.2.

Encoding techniques are evaluated using four ensemble methods: Random Forest (RF) [9], XGBoost [10], LightGBM [11], and CatBoost [12]. To serve as an additional baseline, we also evaluate the performance of the multilayer perceptron (MLP). Classification performance is averaged across six runs of five-fold cross-validation, and Tukey's Honestly Significant Difference (HSD) test is used to identify methods with significantly higher performance. Experimental results show that when including the HCPCS feature the one-hot method consistently performs the worst across all learners. The Hcpcs2Vec embedding outperforms other encoding techniques for four of five learners, and the XGBoost learner performs best overall. The primary contribution of this paper is its diverse evaluation of encoding techniques for high-cardinality variables. Furthermore, we incorporate the latest CMS Medicare 2018 data set and compare results across some of the most popular ensemble learners available.

Section 2 describes related works in the area of encoding high-cardinality features and Medicare fraud detection. Section 3 outlines the Medicare Part B data set, encoding techniques, and experiment methodology. Finally, Section 4 and Section 5 present our results and conclude with areas for future work.

2. Related Work

2.1. High-Cardinality Categorical Features

Moeyersoms and Martens [13] introduce several categorical feature transformations for high-cardinality features on a churn prediction task. A semantic grouping method is used to convert ZIP code attributes to a reduced subset of categories by grouping them into provinces, and the reduced subset is then one-hot encoded. This works when meta data is available to enable the reduction of categories to a sufficiently small subset, but this natural grouping may not always be available. In addition, the authors use three different methods for converting categorical variables to continuous values using target statistics (TS). Techniques are evaluated on C4.5 decision tree, logistic regression, and support vector machine learners. Results conclude that including the high-cardinality features improve performance, but no single technique clearly performs best overall. We include a modified TS encoding technique in this study by leveraging CatBoost's automatic categorical feature handling.

Prokhorenkova et al. [12] introduce CatBoost as an implementation of the gradient boosting machine (GBM) learner that provides automatic handling of categorical variables. The authors suggest that one-hot encodings are not appropriate for high-cardinality features and that alternative TS solutions are liable to introduce target leakage that ultimately leads to overfitting. To mitigate target leakage, the authors suggest computing the TS for the k^{th} sample x_k on a subset of examples $\mathcal{D}_k \subset \mathcal{D}$ that excludes x_k by using ordered permutations of training data. The proposed solution, ordered TS, is shown to outperform existing TS implementations on eight popular benchmark data sets. CatBoost also introduces a modified boosting algorithm, ordered boosting, that is designed to prevent prediction shift. Similar to ordered TS, ordered boosting uses the ordered principle to train models on permutations of data. The authors combine ordered TS and ordered boosting such that the random permutations align, guaranteeing that a target y_i is not used for training model M_i. CatBoost is shown to outperform the LightGBM and XGBoost learners on nine popular benchmarks. In our study, we employ CatBoost learners for Medicare fraud prediction and compare the performance of the ordered TS encodings and ordered boosting to alternative embedding techniques and ensemble learners.

Several research groups have extended word embedding techniques to the biomedical domain and learned new representations for high-dimensional medical concepts, e.g. medications, disease codes, and procedures. De Vine et al. [14] produce medical concept embeddings from free text in clinical records and medical journal abstracts. Skip-gram (SG) Word2Vec models are trained on sequences of medical concepts to learn distributed representations that capture semantic similarities. Choi et al. [15] 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. [14]. The medical concept embeddings from medical claims (MCEMC) are trained on claims data that spans over four million patients between 2005 and 2013. Beam et al. [16] 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. In a previous work [8], we used the SG model to learn distributed representations of HCPCS procedure codes using historical Medicare claims data, i.e. Hcpcs2Vec, and found that it outperforms MCEMC on the Medicare fraud detection problem. Hence, we employ Hcpcs2Vec as our semantic embeddings of choice in this study.

2.2. Medicare Fraud Prediction

The CMS public data repositories have enabled valuable research in predictive modeling for healthcare fraud detection. Bauder and Khoshgoftaar [17] use a series of regression models to estimate Medicare payments and flag fraudulent providers by comparing actual payments to estimated payments. Ko et al. [18] model the variance of service utilization and payments using a linear regression model. Herland et al. [7] explore fraud prediction using three 2012–2015 CMS Medicare data sets, i.e. Part B, Part D, and DMEPOS. Herland et al. do not include HCPCS features directly, but instead encode the information using the Aggregated method that we use as a baseline in this

TABLE 2. DESCRIPTION OF PART B FEATURES

Feature	Description	Type
NPI	Unique provider identification number	Categorical
year	Year of billing activity	Categorical
gender	Provider's gender	Categorical
provider_type	Medical provider's specialty (or practice)	Categorical
hcpcs_code	Procedure/service code	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

study. In another study from Bauder and Khoshgoftaar [19], supervised learners are shown to outperform unsupervised learners and hybrid learners on the Medicare Part B data set. More recently, procedure and prescription drug claims have been used to represent medical providers and improve downstream fraud classification [20]. These studies have incrementally advanced fraud classification efforts, but they do not utilize the high-dimensional categorical features that we are concerned with in this study.

Branting et al. [21] extract features from graph structures of Part B and Part D claims data to improve fraud detection. Providers, prescription drugs, and HCPCS procedure codes are represented as graph nodes that are linked by relations in historical Medicare claims. Behavioral similarity and geospatial co-location features are extracted from the graph and modeled with decision tree learners to classify fraud. Chandola et al. [22] model providers as documents by constructing provider-diagnosis matrices from claims data, and then use Latent Dirichlet Allocation to identify hidden topics that can be used to identify fraudulent providers. Hancock and Khoshgoftaar [23] incorporate the high-cardinality HCPCS procedure code feature into the Medicare Part B classification problem using CatBoost's automatic categorical data handling. CatBoost is shown to significantly outperform an XGBoost learner that is trained without the HCPCS feature, demonstrating the importance of including this feature in future experiments. In another work [8], we presented Hcpcs2Vec for learning distributed representations of HCPCS procedure codes and applied these embeddings to the Medicare Part B data set using the XGBoost learner. Hcpcs2Vec learns low-rank semantic embeddings for procedure codes from historical Medicare data using the Word2Vec algorithm. Continuous-bag-of-words (CBOW) and SG implementations were evaluated against traditional one-hot encodings and open-source medical code embeddings MCEMC [15]. While each of these studies utilize high-cardinality categorical features, they lack the comprehensive comparison of encoding techniques and ensemble learners that we seek to provide in this study.

3. Methods

This section describes the Medicare Part B data set, the different encoding techniques used to represent the HCPCS categorical variable, and the experiment design used to evaluate performance.

3.1. Medicare Part B Data

The 2012–2018 CMS Part B data sets are publicly available in comma delimited format on the CMS website [5]. The data set describes healthcare providers and the services or procedures that they provide to Medicare beneficiaries, and it includes more than 66 million records. Provider data includes demographic data from the National Plan & Provider System (NPPES), data that is collected during provider enrollment and updated regularly. Spending and utilization data is aggregated by the CMS each year on: 1) NPI, 2) Healthcare Common Procedure Coding System (HCPCS) code, and 3) the place of service. A full list of features used in this study are listed in Table 2.

Providers in the Medicare data sets are identified by their NPI. The provider_type attribute consists of 102 unique values that describe a provider's specialty, e.g. Internal Medicine, Nurse Practitioner, and Urology. The HCPCS code attribute includes 7,752 procedure codes that identify specific procedures performed by a provider. For example: G9964 identifies a child wellness visit, V5008 identifies a hearing screening, and M1003 identifies a Tuberculosis test. The remaining numeric attributes describe the provider's utilization for a given HCPCS code on a given year, e.g. the number of services performed, the number of beneficiaries seen, and the average amount charged to Medicare. The NPI and year attributes are used to label records as fraudulent and non-fraudulent when joining with the LEIE data set, but they are not used when training and evaluating classifiers. This leaves a total of eight attributes for classification, three of which are categorical. The provider_type and gender attributes have relatively low cardinality and are therefore one-hot encoded for all experiments in this study. Five encoding techniques are evaluated for encoding the hcpcs_code due to its high cardinality.

Preparing the 2012–2018 Part B data set requires downloading each year of data from the CMS, concatenating the years, and cleaning the data. Column names must be matched and normalized because the CMS has changed attribute names over the years. Records with missing NPI and HCPCS codes are removed. We have also matched and normalized the provider_type attribute names, as these have changed over the years. Rows missing the gender attribute are imputed with a third gender, U, for unknown.

Fraud labels are identified using the publicly available LEIE data set. Excluded individuals are unable to receive

payment from Federal healthcare programs for any services, and must apply for reinstatement once their exclusion period has been satisfied. The LEIE exclusion type attribute is a categorical value that describes the offense and its severity. Following the work by Bauder and Khoshgoftaar [17], a subset of exclusion rules that are most indicative of fraud are selected for labeling Medicare providers. Table 3 lists the exclusion rules used in this study. We use the NPI numbers of excluded individuals that have been convicted under one of these rules to identify fraudulent providers within each of the Medicare data sets. This enables us to correctly label Medicare providers that have been convicted of fraudulent activity. It does not allow us to correctly label all fraudulent Medicare providers, however, as the LEIE data set does contain missing NPI values and there are naturally many borderline corrupt providers who have not been convicted. This adds an unknown degree of class label noise to our Medicare data set that will affect classification performance. We do not treat this class label noise in this study, as our focus is solely on embedding techniques for high-cardinality variables.

TABLE 3. FRAUD RELATED LEIE RULES

Social Security Act	Description
1128(a)(1)	Conviction of program-related crimes
1128(a)(2)	Conviction relating to patient abuse or neglect
1128(a)(3)	Felony conviction relating to health care fraud
1128(b)(4)	License revocation, suspension, or surrender
1128(b)(7)	Fraud, kickbacks, and other prohibited activities
1128(c)(3)(g)(i)	Conviction of second mandatory exclusion offenses
1128(c)(3)(g)(ii)	Conviction of third mandatory exclusion offenses

3.2. Encoding Techniques

The Aggregated method for encoding Medicare data for fraud prediction was first introduced by Herland et al. [7]. This technique aggregates provider records and replaces the high-dimensional categorical attributes with a series of summary statistics. In the Part B data set, we aggregate over the NPI, year, provider_type, and gender attributes, and then drop the HCPCS attribute. When we combine all rows for this provider, we convert each numeric attribute to six summary statistics: minimum, maximum, median, mean, sum, and standard deviation. This transformation is applied to all numeric attributes, and when there exists only one row for a given provider the standard deviation is imputed with 0. While this does reduce the size and dimensionality of the data sets, we suspect that the removal of the HCPCS code and drug name attributes may degrade fraud classification performance as it incurs information loss. We apply the same procedures as Herland et al., but expand to the latest 2018 Medicare data sets, and use the Aggregated technique as a baseline to compare against alternative encoding techniques.

The one-hot encoding technique is arguably the most popular method for encoding categorical variables in machine learning [24]. This method converts a single categorical variable with d distinct values into a binary vector of length d, where the i^{th} entry corresponding to the i^{th} category is 1 and all other entries are 0 [9]. One-hot encoding the HCPCS attribute transforms the categorical feature to a sparse binary vector with 7,752 dimensions. While one-hot encoding is popular and effective for many classification problems, we do not believe it is well suited for the high-dimensional HCPCS feature. The one-hot encoding significantly increases the dimensionality of an already big data set, increasing model complexity and risking performance degradation by the curse of dimensionality [25]. Perhaps even more important, the one-hot encoding represents each categorical value as a mutually exclusive and unrelated category, when in reality there are many relationships and similarities between HCPCS procedure codes.

The CatBoost learner uses the ordered TS strategy introduced in Section 2.1 to represent each category. Ordered TS uses ordered partitions of data to calculate TS in order to ensure that a target is not used for both training and loss evaluation. This helps mitigate overfitting and has been shown to outperform alternative TS implementations on eight popular data sets [12].

LightGBM uses exclusive feature bundling (EFB) to reduce the total number of features used during training [11]. EFB combines non-conflicting features by constructing a graph whose vertices are features and weighted edges denote conflicts with other features. Features are ranked and added to feature bundles, or feature groups, if the conflict between the feature and the bundle's features do not exceed a predetermined threshold λ. In our experiments, LightGBM will apply EFB to the HCPCS categorical variable when we provide it as a one-hot vector. LightGBM also provides a built-in stategy for encoding categorical features that is activated when a categorical feature is integer-encoded or explicitly declared as categorical. In this scenario, LightGBM applies the Fischer technique [26] to find the optimal split over categories. This is achieved by sorting the histogram of categorical features by the accumulated sum of gradients and selecting the optimal split on the sorted histogram. We refer to this encoding technique as the LightGBM encoding in our experiments.

Hcpcs2Vec embeddings are distributed representations of HCPCS procedure codes that we have proposed in a previous work [8]. The embeddings are learned from the Medicare Part B data sets using the Word2Vec SG or CBOW algorithms [27]. 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 [28]. 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. We refer readers to the original paper [8] for complete implementation details. Based on our previous work, we use the SG implementation of Word2Vec with $min\_count = 2$, $iters = 100$, negative sampling with $negative = 5$, window size $L = 10$, and embedding size $d = 75$ in this study. In this previous work, results show that embedding sizes of $d \in \{75, 150, 350\}$ perform

TABLE 4. PART B DATA SUMMARY

Dataset	Years	Records	Positive Count	Positive Ratio
Original Part B Data	2012–2018	66,780,030	40,194	0.0602%
Sampled Part B Data	2012–2018	5,000,000	40,194	0.8039%

approximately the same, and the smaller embedding size of $d = 75$ reduces computational complexity.

3.3. Fraud Classification

Five predictive models are used to evaluate encoding techniques on the Medicare Part B fraud classification task. The first four, and the focus of this study, are bagging and boosting ensemble methods. RF and XGBoost learners are trained using $max\_depth = 8$ and $n\_estimators = 100$. Default parameters are used for the CatBoost and LightGBM learners, i.e. $max\_depth = 6$ and no depth limit for CatBoost and LightGBM, respectively. To serve as an additional baseline for comparison, an MLP learner is also trained using two hidden layers with 62 and 32 neurons per layer, a dropout rate of 0.5, batch normalization, ReLu activation functions, and 50 training epochs. These parameters were selected because they performed best during preliminary grid-search experiments, and any additional hyperparameters not explicitly defined here are left at their default values.

From the 66 million data points in the 2012–2018 Part B data set, only 0.06% are labelled as fraudulent. To address the challenges related to highly imbalanced big data [29], we employ a simple data sampling technique to reduce the size of the majority class [30]. We combine all 40,194 positive samples with a random sample (without replacement) from the non-fraudulent class to create sampled data sets comprised of 5 million records. We select 5 million as the sample size because preliminary results reveal diminishing returns with additional data. As shown in Table 4, this under-sampling increases the size of the positive class to 0.80%. We have shown in previous studies that applying this under-sampling technique to training data improves classification performance on an unsampled test set, despite the differences in class proportion between the train and test set [31]. We refer readers to this related work for additional information on sampling imbalanced training data to maximize performance, as this is outside the scope of this paper. Finally, we use a new random sample of 5 million instances for each iteration of five-fold cross-validation to account for any deviations caused by subsampling the non-fraudulent class.

3.4. Performance Evaluation

Six runs of five-fold cross-validation are completed for each learner and encoding configuration, producing 30 results per set. We record the area under the receiver operating curve (AUC), geometric mean (G-Mean), true positive rate (TPR), and true negative rate (TNR). We use a classification threshold equal to the positive class prior (0.008 = 0.80%) when computing G-Mean, TPR, and TNR results, based on previous studies related to thresholding for imbalanced data [32].

Finally, we use a significance level of $\alpha = 0.05$ to report Tukey's HSD test results for AUC scores and confidence intervals for G-Mean, TPR, and TNR performance. 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 [33]. Encoding techniques are assigned to alphabetic groups based on the statistical difference of AUC means, e.g. group a is significantly different from group b.

4. Results and Discussion

We begin by reviewing the average AUC results of each encoding technique by learner in Table 5. Bold scores indicate the maximum performance for each learner, and the HSD groups listed in parenthesis distinguish results from significantly different distributions. The Hcpcs2Vec encoding technique obtains the highest average AUC score on four of five learners, where both the XGBoost and RF learners see significant improvements when using the Hcpcs2Vec encodings compared to Aggregated and One-hot methods. For the CatBoost and LightGBM learners, the built-in categorical feature encodings perform as well as or better than Hcpcs2Vec. The MLP learner performs approximately the same when using the one-hot and Hcpcs2Vec. The Hcpcs2Vec obtains the best performance with an AUC score of 0.8715 using the XGBoost learner.

When switching from the Aggregated encoding that does not include the HCPCS feature to the one-hot encoding that does, there is no clear change in performance as results vary across learners. For all models, however, the LightGBM, CatBoost, and Hcpcs2Vec encodings outperform the one-hot and Aggregated techniques. This suggests that the inclusion of the HCPCS feature is important to maximizing classification performance, but that the one-hot representation is not appropriate for this high-dimensional problem. Overall, we can conclude that the inclusion of the HCPCS feature significantly improves performance when properly encoded.

The XGBoost and LightGBM models perform best overall with AUC scores of 0.8715 and 0.8486, respectively. The XGBoost, LightGBM, and MLP all outperform CatBoost and RF learners, and the RF learner performs the worst with a maximum AUC of 0.8034. Based on these AUC results, the XGBoost learner is the top performing model.

Table 6 lists the 95% confidence intervals for the G-Mean, TPR, and TNR results with bold scores indicating the maximum average score for a given model and perfor-

TABLE 5. AVERAGE AUC PERFORMANCE AND HSD GROUPS

Encoding	XGBoost	CatBoost	LightGBM	RF	MLP
One-hot	0.8456 (b)	0.7908 (c)	0.8261 (c)	0.7730 (b)	0.8288 (a)
Aggregated	0.8464 (b)	0.8030 (b)	0.7068 (d)	0.7773 (b)	0.7281 (b)
LightGBM	-	-	**0.8486 (a)**	-	-
CatBoost	-	0.8244 (a)	-	-	-
Hcpcs2Vec	**0.8715 (a)**	**0.8249 (a)**	0.8388 (b)	**0.8034 (a)**	**0.8321 (a)**

TABLE 6. CLASSIFICATION CONFIDENCE INTERVALS

Model	HCPCS Encoding	G-Mean	TPR	TNR
CatBoost	One-hot	(0.7212, 0.7235)	(0.7428, 0.7508)	(0.6958, 0.7019)
	Aggregated	(0.7193, 0.7238)	**(0.7585, 0.7683)**	(0.6791, 0.6852)
	CatBoost	(0.7425, 0.7449)	(0.7383, 0.7443)	**(0.7436, 0.7488)**
	Hcpcs2Vec	**(0.7449, 0.7470)**	(0.7513, 0.7583)	(0.7355, 0.7390)
LightGBM	One-hot	(0.7433, 0.7451)	(0.7406, 0.7473)	(0.7421, 0.7471)
	Aggregated	(0.6513, 0.6589)	(0.5866, 0.601)	(0.7165, 0.7298)
	LightGBM	**(0.7629, 0.7652)**	**(0.7606, 0.7667)**	**(0.7616, 0.7674)**
	Hcpcs2Vec	(0.7556, 0.7573)	(0.755, 0.7602)	(0.7528, 0.7579)
MLP	One-hot	(0.7463, 0.7479)	(0.7677, 0.7744)	(0.7213, 0.7268)
	Aggregated	(0.6652, 0.6730)	(0.6800, 0.7363)	(0.6145, 0.6651)
	Hcpcs2Vec	**(0.7506, 0.7521)**	**(0.7714, 0.7788)**	**(0.7252, 0.7315)**
RF	One-hot	(0.6665, 0.6788)	**(0.8413, 0.8668)**	(0.5147, 0.5491)
	Aggregated	(0.6848, 0.6891)	(0.7928, 0.8035)	(0.5884, 0.5943)
	Hcpcs2Vec	**(0.7211, 0.7244)**	(0.7397, 0.7578)	**(0.6881, 0.7093)**
XGB	One-hot	(0.7582, 0.7601)	(0.7439, 0.7508)	(0.7688, 0.7738)
	Aggregated	(0.7063, 0.7119)	(0.5524, 0.5613)	**(0.9025, 0.9037)**
	Hcpcs2Vec	**(0.7823, 0.7841)**	**(0.7658, 0.7727)**	(0.7948, 0.8001)

mance metric. As expected, G-Mean results are correlated with AUC results because we have selected an appropriate classification threshold, i.e. the positive class prior, for computing the confusion matrix. The MLP, RF, and XGBoost learners obtain significantly higher G-Mean scores when using Hcpcs2Vec, and the CatBoost and LightGBM learners perform as well, or better, using their built-in categorical feature encodings. The XGBoost and LightGBM learners perform best overall with G-Mean scores in the ranges of 0.7823–0.7841 and 0.7629–0.7652, respectively. The RF learner performs the worst with a G-mean interval of 0.7211–0.7244. Judging by both AUC and G-Mean, the XGBoost, LightGBM, and MLP learners all significantly outperform the CatBoost and RF learners on this Medicare fraud classification problem.

Both TPR and TNR scores must be taken into consideration when comparing classification performance results for fraud detection. For example, the RF learner obtains the best TPR interval overall at 0.8413–0.8668, but the TNR $\leq$ 0.5491 is unacceptable. Instead, for this classification problem, we would like to maximize the TPR while approximately balancing the TNR. From this perspective, the LightGBM learner performs very well using its internal encoding technique, with a TPR $\geq$ 0.7606 and TNR $\geq$ 0.7616. The XGBoost learner remains the top performer, however, with a TPR $\geq$ 0.7658 and TNR $\geq$ 0.7948. The 0.03 increase in TNR from LightGBM to XGBoost is especially important in this highly imbalanced big data problem, as the negative fraudulent class is the majority class, and a 3% increase in non-fraudulent detection equates to approximately 150,000 providers correctly classified as non-fraudulent.

5. Conclusion

This study evaluated five different encoding techniques for high-cardinality features using ensemble learners. Techniques were evaluated on the latest Medicare Part B fraud classification problem, a big data set with a HCPCS procedure code feature containing 7,752 unique values. Traditional one-hot encodings were compared to an aggregated encoding that excludes the categorical feature. CatBoost and LightGBM's built-in capabilities for high-dimensional categorical variables were then compared to Hcpcs2Vec distributed representations that capture semantic similarities between procedure codes. Statistical results show that the inclusion of the high-cardinality categorical feature significantly improves performance. Specifically for ensemble learners, however, we found that the one-hot encoding provides little-to-no improvement in predictive performance. These results suggest that one-hot encodings are not suitable for high-cardinality features when using ensemble learners. Overall, we found the Hcpcs2Vec embeddings for HCPCS procedure codes to obtain the best performance, and the XGBoost learner performed significantly better than all other learners according to AUC and G-Mean metrics. Future works will extend this study to include additional data sets containing high-cardinality categorical features and will empirically evaluate how the level of cardinality affects performance for different learners.

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] 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

[8] J. M. Johnson and T. M. Khoshgoftaar, "Hcpcs2vec: Healthcare procedure embeddings for medicare fraud prediction," in *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, 2020, pp. 145–152.

[9] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*, 4th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.

[10] 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

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 3149–3157.

[12] 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.

[13] J. Moeyersoms and D. Martens, "Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector," *Decision Support Systems*, vol. 72, pp. 72–81, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923615000275

[14] 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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] J. M. Johnson and T. M. Khoshgoftaar, "Medical provider embeddings for healthcare fraud detection," *SN Computer Science*, vol. 2, no. 4, p. 276, 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00656-y

[21] 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.

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

[23] J. T. Hancock and T. M. Khoshgoftaar, "Gradient boosted decision tree algorithms for medicare fraud detection," *SN Computer Science*, vol. 2, no. 4, p. 268, 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00655-z

[24] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *International journal of computer applications*, vol. 175, no. 4, pp. 7–9, 2017.

[25] L. Chen, *Curse of Dimensionality*. Boston, MA: Springer US, 2009, pp. 545–546.

[26] W. D. Fisher, "On grouping for maximum homogeneity," *Journal of the American Statistical Association*, vol. 53, no. 284, pp. 789–798, 1958. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501479

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

[28] 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

[29] 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

[30] R. A. Bauder and T. M. Khoshgoftaar, "The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data," *Health Information Science and Systems*, vol. 6, no. 1, p. 9, 2018. [Online]. Available: https://doi.org/10.1007/s13755-018-0051-3

[31] 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

[32] J. Johnson and T. M. Khoshgoftaar, "Thresholding strategies for deep learning with highly imbalanced big data," *Deep Learning Applications, Volume 2*, pp. 199–227, 2021. [Online]. Available: https://doi.org/10.1007/978-981-15-6759-9_9

[33] 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