2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)

Deep Learning and Data Sampling with Imbalanced Big Data

Justin M. Johnson, 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 use of deep learning and data sampling on a class-imbalanced *Big Data* problem, i.e. Medicare fraud detection. Medicare offers affordable health insurance to the elderly population and serves more than 15% of the United States population. To increase transparency and help reduce fraud, the *Centers for Medicare and Medicaid Services* (CMS) have made several data sets publicly available for analysis. Our research group has conducted several studies using CMS data and traditional machine learning algorithms (non-deep learning), but challenges associated with severe class imbalance leave room for improvement. These previous studies serve as baselines as we employ deep neural networks with various data-sampling techniques to determine the efficacy of deep learning in addressing class imbalance. Random over-sampling (ROS), random under-sampling (RUS), and combinations of the two (ROS-RUS) are applied to study how varying levels of class imbalance impact model training and performance. Class-wise performance is maximized by identifying optimal decision thresholds, and a strong linear relationship between minority class size and optimal threshold is observed. Results show that ROS significantly outperforms RUS, combining RUS and ROS both maximizes performance and efficiency with a $4\times$ speedup in training time, and the default threshold of 0.5 is never optimal when training data is imbalanced. To the best of our knowledge, this is the first study to provide statistical results comparing ROS, RUS, and ROS-RUS deep learning methods across a range of class distributions. Additional contributions include a unique analysis of thresholding as it relates to the minority class size and state-of-the-art performance on the given fraud detection task.**

***Keywords*—Artificial Neural Networks, Deep Learning, Class Imbalance, Big Data, Data Sampling, Thresholding, Fraud Detection**

1. Introduction

Medicare is a United States (U.S.) healthcare program that offers affordable health insurance to individuals 65 years and older, and other select individuals with permanent disabilities [1]. In 2017, Medicare provided coverage to 58.4 million beneficiaries and exceeded $710 billion in total expenditures [2]. The Federal Bureau of Investigation estimates that fraud accounts for 3–10% of all Medicare billings [3], and the AARP reports that more than $60 billion was lost to fraud in 2017 [4].

One way to save taxpayer money and reduce Medicare premiums is to reduce the total amount of fraud, waste, and abuse within the Medicare program. However, manually auditing claims data for fraudulent activity is very tedious and inefficient. Thanks to the widespread adoption of electronic health record systems [5], machine learning can be used to semi-automate fraud detection with large volumes of data. The *Centers for Medicare and Medicaid Services* (CMS) joined this effort by making certain Medicare data publicly available [6].

This study expands on existing Medicare fraud detection work [7], [8] using the CMS *Medicare Provider Utilization and Payment Data: Physician and Other Supplier Public Use File* [9]. This data set, referred to as *Part B* data hereafter, describes the services provided to Medicare beneficiaries by healthcare professionals. To enable supervised learning, positive fraud labels are mapped to the Part B claims data using the *List of Excluded Individuals and Entities* (LEIE) [10]. The LEIE is maintained by the *Office of Inspector General* (OIG), and it lists providers that are prohibited from participating in Federal healthcare programs. Herland et al. [8] use the five Vs of big data to characterize this as a big data problem: volume, variety, velocity, veracity, and value. The positive class of interest, i.e. fraudulent providers, makes up just 0.03% of the data set. We denote levels of class imbalance in this paper using n_{neg}:n_{pos}, where n_{neg} and n_{pos} correspond to the relative number of samples in the negative and positive class. When training models from such skewed data, standard machine learning algorithms will usually over-predict the majority class [11].

Over the last 10 years, deep learning methods have grown in popularity as they have improved the state-of-the-art in many domains [12]. Recent success can be attributed to an increased availability of data, improvements in hardware and software [13], [14], [15], [16], and various algorithmic breakthroughs that speed up training and improve generalization to new data [17]. Deep learning is a sub-field of machine learning that uses the *artificial neural network* (ANN) with two or more hidden layers to approximate some function f^*, where f^* is commonly used to make

978-1-7281-1337-1/19/$31.00 ©2019 IEEE
DOI 10.1109/IRI.2019.00038

predictions [18]. The ANN, inspired by the biological neural network, is a set of interconnected neurons, or nodes, where connections are weighted and each neuron transforms its input into a single output by applying a non-linear activation function to the sum of its weighted inputs. In a feedforward network, data propagates through the network in a forward pass, each hidden layer receiving its input from the previous layer's output, producing a final output that is dependent on the input data, the choice of activation function, and the weight parameters [19]. Gradient descent optimization adjusts the network's weight parameters in order to minimize the loss function, i.e. the error between expected output and actual output. The deep learning architecture, i.e. *deep neural network* (DNN), achieves its power by composing multiple non-linear layers that learn increasingly complex hierarchical representations. Interested readers can learn more in [18], [19].

This study evaluates the use of DNNs on the Medicare Part B fraud detection task. Class imbalance is addressed using three data-level methods: random over-sampling (ROS), random under-sampling (RUS), and a hybrid ROS-RUS that combines under-sampling and over-sampling. For each method, multiple levels of class imbalance are tested by varying the size of the minority and majority class in the training set. A total of 32 experiments are carried out by applying these methods to both two- and four-hidden-layer networks. Optimal decision thresholds are identified for each experiment on a validation set, and model performance is measured by the average *area under the Receiver Operating Characteristic curve* (ROC AUC) [20] score on a 20% hold-out test set. *Tukey's HSD* (honestly significant difference) [21] test is used to add rigor and estimate the statistical significance of the results.

The remainder of the paper is structured as follows. Section 2 summarizes related works in areas of CMS Medicare data, fraud detection, and deep learning with class-imbalanced data. Section 3 describes the experimental framework, data sets, data sampling methods, and evaluation criteria. Results are presented in Section 4, and Section 5 concludes the study with areas for future works.

2. Related Work

2.1. Medicare Fraud Detection

Our research group has performed extensive research in the area of detecting anomalous provider behavior with CMS Medicare data. Bauder and Khoshgoftaar [22] proposed an outlier detection method that uses Bayesian inference to identify outliers, and successfully validated their model using claims data of a known Florida provider that was under criminal investigation for excessive billing. This experiment used a subset of 2012—2014 Medicare Part B data that included dermatology and optometry claims from Florida office clinics. In another study, Bauder and Khoshgoftaar [23] use a subset of the 2012–2013 Medicare Part B data, i.e. Florida claims only, to model expected amounts paid to providers for services rendered to patients. The authors flag potential fraudulent providers by comparing actual payment amount deviations to the expected payment amounts using five different regression models. Another paper by Bauder et al. [24] uses a Naive Bayes classifier to predict provider specialty types, and then flag providers that are practicing outside their expected specialty type as fraudulent. This study also used a Florida-only subset of 2013 Medicare Part B claims data, but unlike the previous work this experiment included all 82 provider types. The authors conclude that specialties with unique billing procedures, e.g. audiologist or chiropractic, are able to be classified with high precision and recall. Herland et al. [25] expanded upon the work from [24] by incorporating 2014 Medicare Part B data and real-world fraud labels defined by the LEIE data set. A Naive Bayes classifier is used to predict a provider's specialty type, and providers that are misclassified are labeled as potentially fraudulent. The authors find that grouping similar specialty types, e.g. Ophthalmology and Optometry, improves overall performance. Bauder and Khoshgoftaar [7] merge the 2012–2015 Medicare Part B data sets, label fraudulent providers using the LEIE data set, and compare multiple traditional machine learning classifiers. Class imbalance is addressed with RUS, and various class distributions are generated to identify the optimal imbalance ratio for training. The C4.5 decision tree and logistic regression (LR) learners significantly outperform the support vector machine (SVM), and the 80:20 class distribution is shown to outperform 50:50, 65:35, and 75:25 distributions. These studies jointly show that the Medicare Part B claims data contains sufficient variability to detect bad actors and that the LEIE data set can be reliably used for ground truth fraud labels.

Our study is most closely related to the work performed by Herland et al. in [8]. The authors use the Part B, Part D [26], and DMEPOS [27] claims data independently to perform cross-validation with LR, Random Forest (RF), and Gradient Boosted Tree (GBT) learners. They also create a fourth data set that combines all three data sets to determine if learners should be trained on each data set independently, or on all available data. Results show that the combined and Part B data sets score significantly better on ROC AUC than the other data sets, and the LR learner outperforms the GBT and RF learners with a max AUC of 0.816.

A few key research groups have used the CMS Medicare and LEIE data to identifying patterns, anomalies, and potentially fraudulent providers. Feldman and Chawla [28] looked for anomalies in the relationship between medical school training and the procedures that physicians perform in practice by linking 2012 Medicare Part B data with provider medical school data obtained through the CMS physician compare data set [29]. Significant procedures for each school were used to evaluate school similarities and present a geographical analysis of procedure charges and payment distributions. Ko et al. [30] used the 2012 CMS data to analyze the variability of service utilization and payments. The authors found that the number of patient visits is strongly correlated with Medicare reimbursement,

and concluded that there is a possible 9% savings within the field of Urology alone. Chandola et al. [31] use claims data and fraud labels from the Texas Office of Inspector General's exclusion database to detect anomalies. They confirm the importance of including provider specialty types in fraud detection, showing that the inclusion of specialty attributes increases AUC scores from 0.716 to 0.814. Branting et al. [32] propose a graph-based method for estimating healthcare fraud risk within the 2012–2014 CMS PUF and LEIE data sets. The authors leverage the NPPES [33] registry to look up NPI numbers that are missing from the LEIE database, increasing their total fraudulent provider count to 12,000. They combine these fraudulent providers with a subset of 12,000 non-fraudulent providers and employ a J48 decision tree learner to classify fraud with a mean AUC of 0.96. This high AUC score is misleading because the class-balanced data set is not representative of the underlying Medicare claims data.

2.2. Deep Learning with Class Imbalance

Despite advances in deep learning and its increasing popularity, many researchers find the subject of deep learning with class-imbalanced data to be understudied [34], [35], [36], [37], [38], [39]. Anand et al. [40] studied the effects of class imbalance on the backpropagation algorithm in shallow networks and show how the optimization process is dominated by the majority class. Network weight updates being driven primarily by the majority class causes the majority class error to reduce rapidly, but this is usually at the expense of increasing minority class error. In a recent paper [41], we surveyed 15 deep learning methods for addressing class imbalance that address this underlying problem.

Hensman and Masko [42] explored the effects of ROS on class imbalanced image data by generating 10 class-imbalanced distributions from the *CIFAR-10* [43] data set. The ROS method randomly duplicates minority class examples until all classes are balanced, where any class whose size is less than that of the largest is considered to be a minority. The authors compare the results achieved with over-sampling to the results obtained on the original balanced distribution, and show that over-sampling minority classes until all classes are balanced restores performance and achieves results comparable to those achieved on the original balanced data set. Buda et al. [39] presented similar results when they compared ROS with RUS. The ROS and RUS methods used by Buda et al. over-sample the minority and under-sample the majority until classes have an equal number of samples. The authors showed that ROS consistently outperforms both RUS and two-phase learning. In another set of experiments by Dong et al. [44], over-sampling is shown to consistently outperform under-sampling on the CelebA image data set [45].

These related works all suggest training DNNs with class balanced data by over-sampling the minority class. These studies do not consider the added complexity of big data and class rarity. We believe that ROS will become inefficient as the size of the majority class and level of imbalance increases. Hence, we consider combining RUS with ROS to exploit the advantages of each. Additionally, we extend related works by reporting performance over a range of class imbalance levels. To the best of our knowledge, this is the first study to compare ROS, RUS, and ROS-RUS over a range of class ratios with deep neural networks and class-imbalanced data. Furthermore, we believe that related works have not emphasized the importance of thresholding when training data is imbalanced. We address this research gap by providing a unique analysis on the relationship between the minority class size and the optimal decision threshold.

3. Methodology

DNN models are evaluated on the Medicare Part B data set by fitting models on a 80% training set and reporting performance on a 20% test set. From the training set, 10% is used to validate hyperparameters and calculate optimal decision thresholds. Neural networks are implemented using the Keras [15] open-source deep learning library with its default backend, i.e. TensorFlow [13].

3.1. Data Sets

This study uses the 2012–2016 Medicare Part B data sets provided by CMS [9]. The Medicare Part B claims data describes the services and procedures that healthcare professionals provide to Medicare's Fee-For-Service beneficiaries. Records contain various provider-level attributes, including a unique 10-digit identification number for providers, i.e. the National Provider Identifier (NPI) [46], and the provider specialty type. Other attributes describe the provider's activity within Medicare over a single year, e.g. procedures performed, average charges submitted to Medicare, and average payments by Medicare. Procedures rendered are encoded using the Healthcare Common Procedures Coding System (HCPCS) [47]. CMS releases data annually and aggregates the data over: (1) provider NPI, (2) HCPCS code, and (3) place of service. This produces one record for each provider, HCPCS code, and place of service combination over a given year.

The LEIE data set lists providers that are prohibited from practicing and is used to label providers within the Medicare Part B data set as fraudulent or non-fraudulent. The OIG has the authority to exclude providers from Federally funded healthcare programs for a variety of reasons. Following the work by Bauder and Khoshgoftaar [23], a subset of exclusion types that are indicative of fraud are used to label Medicare providers. We label providers in the Medicare data as fraudulent by matching on NPI numbers, and we consider all claims prior to the provider's exclusion date to be fraudulent.

For each year, records are grouped by NPI and provider type. Each group is converted into a single record of summary statistics, i.e. minimum, maximum, sum, median, mean, and standard deviation. Categorical features are one-hot encoded, and stratified random sampling is used to set aside a 20% test set. A min-max scaler is fit to the training

data and used to transform the attributes of the training and test sets to the range $[0, 1]$. Train and test set details are illustrated in Table 1.

TABLE 1. TRAINING AND TEST DATA SETS' DETAILS

Data Set	Total Samples	Fraudulent Samples	% Fraudulent
Training Data	3,753,896	1206	0.032%
Test Data	938,474	302	0.032%

Interested readers can learn more about these data sets and pre-processing details in the original paper by Herland et al. [8].

3.2. Baseline Models

Hyperparameters are defined through a random search procedure by averaging validation results over 10 runs. All models are trained using the Adam optimizer [48] with mini-batch sizes of 256 and default moment estimate decay rates. The Rectified Linear Unit (ReLU) activation function is used in all hidden layer neurons, and the sigmoid activation function is used at the output layer.

Validation results show that two hidden layers composed of 32 neurons provides sufficient capacity to overfit the training data. We apply batch normalization [49] before hidden-layer activations to speed up training and improve performance on the validation set. Dropout is applied after hidden-layer activations to further reduce overfitting [50]. This baseline architecture is detailed in Table 2. We extend this model to four hidden layers, following the same pattern, and run all experiments on both architectures to determine how depth affects performance.

TABLE 2. BASELINE ARCHITECTURE

Layer Type	# of Neurons	# of Parameters
Input	125	0
Dense	32	4032
Batch Normalization	32	128
ReLU Activation	32	0
Dropout $P = 0.5$	32	0
Dense	32	1056
Batch Normalization	32	128
ReLU activation	32	0
Dropout $P = 0.5$	32	0
Dense	1	33
Sigmoid activation	1	0

3.3. Data-Level Methods

Data-level methods explored in this paper include ROS, RUS, and combinations of ROS and RUS (ROS-RUS). Sampling rates are adjusted to create distributions with varying levels of class imbalance to better understand how class imbalance levels affect training and performance. These distributions are listed in Table 3. The first row describes the training data prior to data sampling, and the remaining rows provide the size of the positive and negative classes after applying data sampling. $N_{train} = n_{neg} + n_{pos}$ denotes the total number of samples in the training set.

TABLE 3. ROS, RUS, AND ROS-RUS EXPERIMENTS

Method	n_{neg}	n_{pos}	N_{train}	n_{neg}:n_{pos}
———	3,377,421	1,085	3,378,506	99.97:0.03
RUS-1	107,402	1,085	108,487	99:1
RUS-2	4,390	1,085	5,475	80:20
RUS-3	1,620	1,085	2,705	60:40
RUS-4	1,085	1,085	2,170	50:50
RUS-5	710	1,085	1,795	40:60
ROS-1	3,377,421	33,635	3,411,046	99:1
ROS-2	3,377,421	844,130	4,221,551	80:20
ROS-3	3,377,421	2,251,375	5,628,796	60:40
ROS-4	3,377,421	3,377,421	6,754,842	50:50
ROS-5	3,377,421	5,064,780	8,442,201	40:60
ROS-RUS-1	1,688,710	1,688,710	3,377,420	50:50
ROS-RUS-2	844,355	844,355	1,688,710	50:50
ROS-RUS-3	337,742	337,742	675,484	50:50

The RUS procedure randomly samples from the majority class without replacement. Under-sampling drastically decreases the size of the training set, which consequently allows for faster training and turnaround times. To create a class ratio of 99:1, which is still highly imbalanced, RUS-1 combines the positive group with a negative class sample that is just 3.18% of the original negative group. This reduces the size of the negative class training set from 3,377,421 to just 107,402. We suspect that RUS will perform poorly due to an under-represented negative class.

The ROS method randomly duplicates minority class samples until a desired level of class imbalance is achieved. Since there are many more non-fraud cases than there are fraud, the fraud cases must be over-sampled at high rates to balance out class distributions. For example, creating a 50:50 class balanced training set with ROS requires sampling the minority class at a rate of 3112%. This increases the size of the minority class from 1085 samples up to 3,377,421 and approximately doubles the size of the training set. The balanced distributions should improve performance, but at the cost of increased training times. This is especially exacerbated by the presence of big data and class rarity.

Finally, we combine ROS and RUS (ROS-RUS) to produce three class balanced training sets. We reduce the majority group by 90%, 75%, and 50% while simultaneously over-sampling the minority group until classes are balanced. Higher reduction rates decrease the size of the training set and improve efficiency, while lower reduction rates improve the representation of the majority group. As shown in Table 3, the largest ROS-RUS training set has 3,377,420 samples, which is still smaller than the original training set.

3.4. Performance Metrics

This study utilizes multiple complementary evaluation metrics to provide a clear understanding of model performance and class-wise score trade-offs [51]. We report the *True Positive Rate* (TPR), *True Negative Rate* (TNR), and *Geometric Mean* (G-Mean) [19] scores on all experiments. These performance metrics are dependent on the decision

threshold that is used to assign labels to output probability estimates. We find that a default threshold of 0.5 causes baseline models to always predict the non-fraudulent label. Therefore, we rely on the threshold-agnostic ROC AUC score to determine how well a model can discriminate between the positive and negative class. If a model produces a reasonable ROC AUC score, then there must exist a decision threshold that will yield equally-reasonable TPR and TNR scores. Validation results are used to calculate the optimal decision threshold for each method, and this method is then applied to the final model on the test set. We find that the level of class imbalance within the training data has a significant impact on the optimal decision threshold, and we believe that selecting an optimal decision threshold is a critical component when learning from class imbalance data.

3.5. Threshold Moving

To improve overall accuracy and better illustrate the efficacy of DNNs in detecting Medicare fraud, we calculate optimal decision thresholds for each method. We prefer a high TPR over a high TNR because detecting fraud is more important than detecting non-fraud. Additionally, we wish to approximately balance the TPR and TNR rates in order to maximize the model's total predictive power. Following these goals, we identify optimal decision thresholds by identifying the threshold that maximizes the G-Mean score on the validation set under the constraint that $TPR > TNR$. The TPR, TNR, and G-Mean results presented in this study are dependent on this threshold selection procedure, and the bias towards the positive class can be increased or decreased by defining a new threshold selection procedure.

4. Results and Discussion

We present the average ROC AUC, TPR, TNR, and G-Mean scores averaged over 30 iterations. Since the TPR, TNR, and G-Mean scores are dependent on the details of the thresholding method, ROC AUC is used as the primary metric for comparing methods. In each experiment, -2 or -4 are appended to method names to distinguish between models containing two and four hidden layers, respectively. Tukey's HSD test and training time analysis are used to select the best performing model.

4.1. Baseline Model Performance

Table 4 lists the results of the baseline DNNs defined in Section 3.2. To better establish a firm baseline for the 2012 - 2016 Medicare Part B fraud detection problem, we have included the scores of three traditional machine learning algorithms in Table 5. No class imbalance techniques are applied to these baseline models.

The DNN Baseline-2 performed second best with an average ROC AUC of 0.8058, runner up to the LR learner with an average ROC AUC of 0.8076. We stress the importance of the decision threshold, noting that it is not until the threshold is decreased to 0.0002 and 0.0003 that the baseline DNNs achieve reasonable TPR and TNR. We also observe that the optimal decision threshold is approximately the same as the minority class size, i.e. 0.03%. This relationship is investigated further in Section 4.7.

TABLE 4. AVERAGE BASELINE DNN RESULTS (30 RUNS)

Method	Decision Threshold	ROC AUC	TPR	TNR	G-Mean
Baseline-2	0.0002	**0.8058**	0.8280	0.6099	0.7088
Baseline-4	0.0003	0.8018	0.7488	0.7135	0.7301

TABLE 5. AVERAGE RESULTS OF TRADITIONAL LEARNERS (10 RUNS)

	Logistic Regression	Random Forest	Gradient Boosted Tree
Avg ROC AUC	**0.8076**	0.7937	0.7990

4.2. RUS Performance

RUS results are listed in Table 6. RUS-1-2, with a 99:1 class distribution, scored the highest of the RUS methods and outperformed all baseline learners with an average ROC AUC of 0.8124 and G-Mean of 0.7383. RUS-2-2, with an 80:20 class distribution, did not perform as well as RUS-1-2, but it does outperform the baseline DNN models. Results show that performance decreases as the size of the negative class decreases.

TABLE 6. AVERAGE RUS RESULTS (30 RUNS)

Method	n_{neg}:n_{pos}	Decision Threshold	ROC AUC	TPR	TNR	G-Mean
RUS-1-2	99:1	0.0110	**0.8124**	0.7807	0.6987	0.7383
RUS-1-4		0.0145	0.8040	0.7581	0.7002	0.7265
RUS-2-2	80:20	0.2680	0.8076	0.7521	0.7163	0.7338
RUS-2-4		0.3520	0.7920	0.7674	0.6853	0.7228
RUS-3-2	60:40	0.4200	0.8043	0.7783	0.6700	0.7212
RUS-3-4		0.5370	0.7907	0.7978	0.6288	0.7021
RUS-4-2	50:50	0.4970	0.8027	0.7864	0.6601	0.7195
RUS-4-4		0.6078	0.7913	0.7778	0.6422	0.6966
RUS-5-2	40:60	0.5730	0.7994	0.7802	0.6588	0.7154
RUS-5-4		0.7060	0.7802	0.7226	0.6462	0.6412

Two additional RUS experiments were conducted to determine if increasing the majority class size will continue to increase performance. Class ratios of 99.5:0.5 and 99.9:0.1 were evaluated and results show that increasing the size of the majority class beyond 99% does not improve performance beyond that of ROS-1-2. This suggests that both the imbalance level and the representation of the majority class are important factors in model performance.

4.3. ROS Performance

ROS results are listed in Table 7. Method ROS-4-2, with a 50:50 class distribution, performed the best with an average ROC AUC of 0.8505 and average G-Mean of 0.7692. ROS-1-4, with the highest level of class imbalance

in its training set, performed the worst with an average ROC AUC of 0.8325, but still outperformed all RUS methods from Table 6. Similar to related works [39], [42], [44], ROS outperforms RUS in all experiments and over-sampling works best when imbalance is eliminated from the training data.

TABLE 7. AVERAGE ROS RESULTS (30 RUNS)

Method	n_{neg}:n_{pos}	Decision Threshold	ROC AUC	TPR	TNR	G-Mean
ROS-1-2	99:1	0.0110	0.8383	0.8572	0.6334	0.7338
ROS-1-4		0.0130	0.8325	0.8064	0.6857	0.7372
ROS-2-2	80:20	0.2410	0.8484	0.8282	0.6926	0.7549
ROS-2-4		0.3000	0.8440	0.8497	0.6165	0.7109
ROS-3-2	60:40	0.4080	0.8454	0.8056	0.7198	0.7582
ROS-3-4		0.4370	0.8438	0.8163	0.6820	0.7385
ROS-4-2	50:50	0.4530	**0.8505**	0.8084	0.7324	0.7692
ROS-4-4		0.4740	0.8389	0.8066	0.6861	0.7365
ROS-5-2	40:60	0.5630	0.8503	0.8163	0.7272	0.7701
ROS-5-4		0.5950	0.8423	0.8086	0.7023	0.7508

4.4. ROS-RUS Performance

ROS-RUS results are listed in Table 8. The Neg. Class Reduction column denotes the amount of the majority class that is removed prior to applying over-sampling. For example, ROS-RUS-2 creates a 50:50 class distribution in the training data by removing 75% of the negative class and then over-sampling the positive class until classes are balanced. All three ROS-RUS methods outperform the RUS learners and perform similarly to the best ROS method. ROS-RUS-2-2 performs the best across all data-level methods with an average ROC AUC of 0.8509 and G-Mean of 0.7710, and trains approximately $4\times$ faster than ROS-4. ROS-RUS-3 has the highest reduction rate and performs the worst of all the ROS-RUS methods.

Similar to ROS results, ROS-RUS results support training DNN models with class-balanced data. Results also suggest that training models with a sufficiently large random sample of the majority class performs as well as the full majority class. This makes ROS-RUS ideal for big data problems, as it maximizes both performance and efficiency.

TABLE 8. AVERAGE ROS-RUS RESULTS (30 RUNS)

Method	Neg. Class Reduction	Decision Threshold	ROC AUC	TPR	TNR	G-Mean
ROS-RUS-1-2	50%	0.5090	0.8500	0.8029	0.7354	0.7665
ROS-RUS-1-4		0.4820	0.8454	0.8064	0.7189	0.7597
ROS-RUS-2-2	75%	0.5218	**0.8509**	0.7876	0.7553	0.7710
ROS-RUS-2-4		0.5140	0.8443	0.7992	0.7175	0.7526
ROS-RUS-3-2	90%	0.4850	0.8477	0.8104	0.7209	0.7625
ROS-RUS-3-4		0.5020	0.8425	0.8063	0.7161	0.7585

4.5. Statistical Analysis

AUC scores are used to select the best methods from each group for further analysis, i.e. RUS-1-2, ROS-4-2, and ROS-RUS-2-2. Tukey's HSD results (Table 9) groups these methods into three distinct categories based on AUC performance, i.e. a, b, and c. These groups are defined by the pair-wise statistical differences between method AUC scores, and each group is statistically different from the other with a confidence of at least 95%. ROS-RUS-2-2 and ROS-4-2 in group *a* obtain significantly higher scores than all other methods, with mean AUC scores of 0.8509 and 0.8505, respectively. RUS-1-2, placed in group *b* with an average AUC score of 0.8124, performs significantly better than the baseline learner.

TABLE 9. TUKEY'S HSD TEST RESULTS (AUC)

Method	Group	AUC	sd	Min	Max
ROS-RUS-2-2	a	0.8509	0.0038	0.8433	0.8591
ROS-4-2	a	0.8505	0.0038	0.8430	0.8594
RUS-1-2	b	0.8124	0.0030	0.8045	0.8170
Baseline-2	c	0.8058	0.0013	0.8029	0.8080

4.6. Training Time Analysis

Table 10 lists the average time to complete one training epoch for each method, where averages are computed across 50 epochs. Various factors influence training time, including the size of the training set (N_{train}), network topology, activation functions, and other hyperparameters. Since all methods were trained for exactly 50 epochs, we can compare methods directly using the time to train one epoch. Taking training times into consideration, we prefer ROS-RUS-2-2 over ROS-4-2 because AUC scores are statistically the same and ROS-RUS-2-2 trains approximately $4\times$ faster. RUS-1-2 sees more than a $30\times$ speedup in training when compared to baseline methods, making it a good choice for preliminary experimentation and hyperparameter tuning.

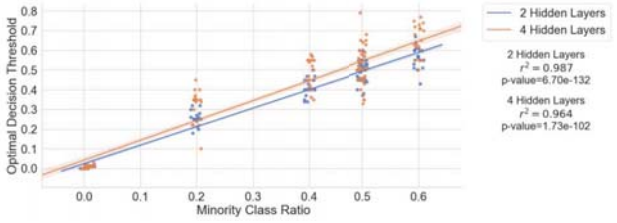
TABLE 10. AVERAGE TRAINING TIME PER EPOCH

Method	Time (s)	sd	N_{train}
RUS-1-2	1.9213	0.0294	108,487
ROS-RUS-2-2	31.0784	0.7683	1,688,710
Baseline-2	63.0775	1.8798	3,378,506
ROS-4-2	128.2847	3.8514	6,754,842

4.7. Analysis of Decision Thresholds

When training networks comprised of two hidden layers with the cross-entropy loss, the learned decision boundary appears to fall near the minority class distribution size. For example, Baseline-2 has a minority class ratio of 0.0003, and the average optimal decision boundary calculated on the trained model is 0.0002. On the other hand, ROS-4-2 has a minority class ratio near 0.5, and the average optimal decision boundary was found to be 0.4530. We plot the minority class size against the optimal decision threshold for all experiments and fit a linear model to the data points using *Ordinary Least Squares* [52]. Results indicate a strong linear relationship with r^2 = 0.987 and p-value = 6.70e-132

Figure 1. Minority Class Ratio vs Optimal Decision Threshold (CE Loss)



for the two-hidden-layer networks. Plots for the two- and four-hidden-layer networks are illustrated in Figure 1 with 0.01 horizontal jitter and 95% confidence interval bands.

5. Conclusion

This study evaluates the use of deep neural networks and data sampling for detecting fraud in severely class-imbalanced data, i.e. 2012–2016 Medicare Part B claims data. The Medicare program provides affordable healthcare to more than 60 million U.S. residents, and it is estimated that between $20 and $70 billion is lost each year to fraud, waste and abuse. We compare three data-level methods for addressing class imbalance and perform statistical analysis to estimate the significance of their results. In doing so, we achieve the highest known ROC AUC scores to date on the given CMS/LEIE data set. ROS and ROS-RUS outperform RUS and baseline models with average AUC scores of 0.8505 and 0.8509, respectively. Results show that the default decision threshold of 0.5 is never optimal when training data is imbalanced, and we suggest that thresholding always be used to optimize class-wise performance when classes are imbalanced. With $4\times$ faster training times compared to ROS, we conclude that deep learning with ROS-RUS is the preferred method for detecting fraud within the CMS Medicare data sets. Furthermore, we recommend the use of RUS for preliminary experiments and hyperparameter tuning, as it significantly outperforms baseline models and achieves up to a $30\times$ improvement in training times.

There are several opportunities for future work in this area. Methods for calculating RUS rates that maximize both efficiency and performance will prove useful to big data modeling. We encourage future work in order to investigate the relationship between the minority class size and the optimal decision threshold across a range of data sets, domains, and architectures. Regarding the Medicare fraud problem, future work should explore improving data quality by leveraging the NPPES registry to look up missing NPI numbers. Additionally, we believe that replacing the one-hot specialty type features with a dense semantic embedding that captures specialty overlap will improve results.

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. (2018) Trustees report & trust funds. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/ReportsTrustFunds/index.html

[3] 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.

[4] J. Eaton. (2018) Medicare under assault from fraudsters. [Online]. Available: https://www.aarp.org/money/scams-fraud/info-2018/medicare-scams-fraud-identity-theft.html

[5] The Office of the National Coordinator for Health Information Technology. Office-based physician electronic health record adoption. [Online]. Available: https://dashboard.healthit.gov/quickstats/quickstats.php

[6] *Medicare Fraud & Abuse: Prevention, Detection, and Reporting*. Centers for Medicare & Medicaid Services, 2017. [Online]. Available: https://www.cms.gov/Outreach-and-Education/Medicare-Learning-Network-MLN/MLNProducts/Downloads/Fraud_and_Abuse.pdf

[7] R. A. Bauder and T. M. Khoshgoftaar, "The detection of medicare fraud using machine learning methods with excluded provider labels," in *FLAIRS Conference*, 2018.

[8] 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

[9] Centers For Medicare & Medicaid Services. (2018) Medicare provider utilization and payment data: Physician and other supplier. [Online]. Available: https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/medicare-provider-charge-data/physician-and-other-supplier.html

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

[11] 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: ACM, 2007, pp. 935–942.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 EP –, 05 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/

[14] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[15] F. Chollet *et al.*, "Keras," https://keras.io, 2015. [Online]. Available: https://keras.io

[16] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," 10 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: The MIT Press, 2016.

[19] 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.

[20] 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.

[21] 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

[22] R. A. Bauder and T. M. Khoshgoftaar, "A probabilistic programming approach for outlier detection in healthcare claims," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2016, pp. 347–354.

[23] 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.

[24] 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)*, Nov 2016, pp. 784–790.

[25] 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.

[26] Centers For Medicare & Medicaid Services. (2018) Medicare provider utilization and payment data: Part d prescriber. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Part-D-Prescriber.html

[27] ——. (2018) Medicare provider utilization and payment data: Referring durable medical equipment, prosthetics, orthotics and supplies. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/DME.html

[28] K. Feldman and N. V. Chawla, "Does medical school training relate to practice? evidence from big data," in *Big data*, 2015.

[29] Centers for Medicare & Medicaid Services. (2019) Physician compare datasets. [Online]. Available: https://data.medicare.gov/data/physician-compare

[30] 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.

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

[32] 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.

[33] National Plan & Provider Enumeration System. (2019) Nppes npi registry. [Online]. Available: https://npiregistry.cms.hhs.gov/registry/

[34] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 4368–4374.

[35] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 3573–3587, 2018.

[36] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 3713–3717.

[37] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y. Lu, S. Chen, and M. Shyu, "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, vol. 00, Apr 2018, pp. 112–117. [Online]. Available: doi.ieeecomputersociety.org/10.1109/MIPR.2018.00027

[38] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5375–5384.

[39] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249 – 259, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608018302107

[40] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 962–969, Nov 1993.

[41] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, Mar 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5

[42] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," 2015, kTH, School of Computer Science and Communication (CSC).

[43] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[44] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[45] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[46] Centers for Medicare & Medicaid Services. (2019) National provider identifier standard (npi). [Online]. Available: https://www.cms.gov/Regulations-and-Guidance/Administrative-Simplification/NationalProvIdentStand/

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

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167

[50] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[51] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, Nov 2009, pp. 59–66.

[52] B. Zdaniuk, *Ordinary Least-Squares (OLS) Model*. Dordrecht: Springer Netherlands, 2014, pp. 4515–4517.