Journal of Big Data

RESEARCH Open Access

Investigating the relationship between time and predictive model maintenance

Joffrey L. Leevy[1*], Taghi M. Khoshgoftaar[1], Richard A. Bauder[1] and Naeem Seliya[2]

*Correspondence:
jleevy2017@fau.edu
[1] Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA
Full list of author information is available at the end of the article

Abstract

A majority of predictive models should be updated regularly, since the most recent data associated with the model may have a different distribution from that of the original training data. This difference may be critical enough to impact the effectiveness of the machine learning model. In our paper, we investigate the relationship between time and predictive model maintenance. Our work incorporates severely imbalanced big data from three Medicare datasets, namely Part D, DMEPOS, and Combined, that have been used in several fraud detection studies. We build training datasets from year-groupings of 2013, 2014, 2015, 2013–2014, 2014–2015, and 2013–2015. Our test datasets are built from the 2016 data. To mitigate some of the adverse effects from the severe class imbalance in these datasets, the performance of five class ratios obtained by *Random Undersampling* and five learners is evaluated by the *Area Under the Receiver Operating Characteristic Curve* metric. The models producing the best values are as follows: *Logistic Regression* with the 2015 year-grouping at a 99:1 class ratio (Part D); *Random Forest* with the 2014-2015 year-grouping at a 75:25 class ratio (DMEPOS); and *Logistic Regression* with the full 2015 year-grouping (Combined). Our experimental results show that the largest training dataset (year-grouping 2013–2015) was not among the selected choices, which indicates that the 2013 data may be outdated. Moreover, we note that because the best model is different for Part D, DMEPOS, and Combined, this suggests that these three datasets may actually be sub-domains requiring unique models within the Medicare fraud detection domain.

Keywords: Model maintenance, Big data, Class imbalance, Machine learning, Undersampling

Introduction

Regularly updating a predictive model is necessary to ensure its effectiveness over time. Some machine learning practitioners, who believe data distributions are temporally static, may assume that maintenance of such models is not warranted [1]. However, this assumption is usually invalid. As an example, a model derived 30 years ago for predicting the effects of anxiety on US teenagers will certainly have to be updated to allow for the influence of social media. The temporal variation of data distributions is denoted by several terms, such as concept drift [2–5], dataset shift [6], and non-stationarity [7]. Figure 1 [8] illustrates the idea that concept drift is a divergence of model and data decision boundaries which leads to a loss of predictability. We note

Springer Open

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

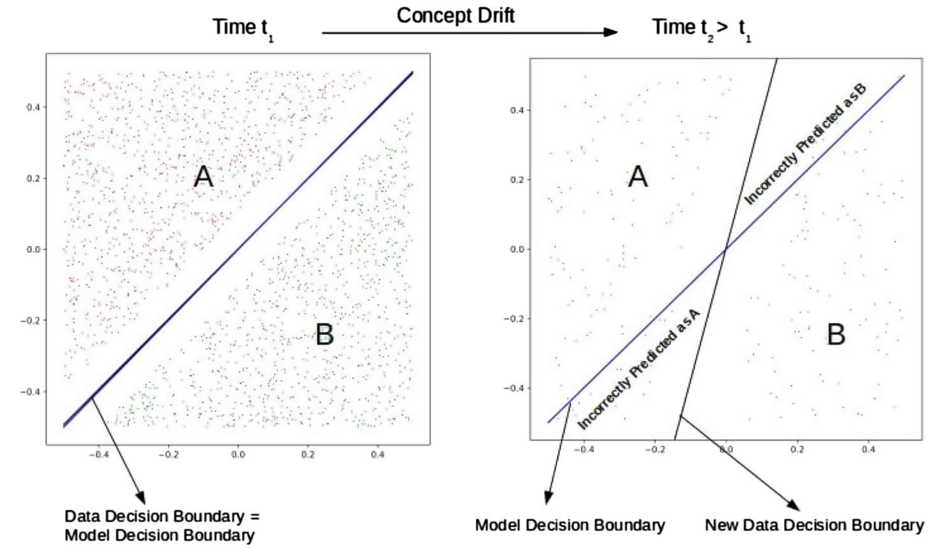

Fig. 1 Concept drift illustration [8]

that our paper only focuses on predictive model maintenance vis-à-vis the dynamic nature of data distributions. Using new or improved machine learning algorithms to facilitate this maintenance is outside the scope of this paper.

We adopt an existing model for detecting Medicare fraud as our frame of reference [9]. This model was selected out of several possible combinations constructed from processed Medicare datasets (Part B, Part D, *Durable Medical Equipment, Prosthetics, Orthotics and Supplies* (DMEPOS), and Combined), and employs the *Logistic Regression* (LR) learner after applying *Random Undersampling* (RUS) with a 90:10 majority-to-minority class ratio. The Combined dataset was created from a join operation of the other three processed datasets. To determine whether or not a physician has committed fraud, we consult the *List of Excluded Individu- als/Entities* (LEIE) to obtain fraud labels which are subsequently mapped to the Medicare datasets. Our work investigates the effect of using training datasets of various year-groupings, where a grouping refers to one or more years' worth of collected data. Training datasets are created from the following year-groupings: 2013, 2014, 2015, 2013–2014, 2014–2015, and 2013–2015. The test datasets are created from 2016 data. Both training and test datasets are constructed from Part D, DMEPOS, and Combined, with the full datasets characterized as highly imbalanced big data.

Although there is no universally accepted definition of big data, data scientists often refer to the six V's: volume, variety, velocity, variability, value, and veracity [10]. Volume, the best-known characteristic of big data, is associated with the amount of data produced by an entity. Variety encompasses the handling of structured, semi-structured, and unstructured data. Velocity considers the speed at which data is manufactured, issued, and handled. Variability pertains to data fluctuations. Value is frequently designated as a critical attribute with respect to effective decision-making. Veracity involves the fidelity of data.

If a dataset has distinct majority and minority classes, e.g., normal and fraudulent transactions for an international bank, the data can be regarded as class-imbalanced. The imbalance is often associated with binary classification, a framework of only a majority class and a minority class, in contrast to a multi-class classification [11] framework of more than two classes. With binary classification, the minority (positive) class comprises a smaller portion of the dataset and is usually the class of interest in real-world problems [12, 13]. For instance, defective hard drives advancing along the production line of a factory constitute the class of interest, while non-defective hard drives make up the majority (negative) class [14]. According to one school of thought, high or severe class imbalance can be expressed in terms of a majority-to-minority ratio between 100:1 and 10,000:1 [15].

From a classification viewpoint, machine learning algorithms are usually more effective than traditional statistical techniques [16–18]. However, these algorithms may be unable to distinguish between majority and minority classes if the dataset is highly imbalanced. As a result, practically every instance may be labeled as the majority (negative) class, and performance metric values based on the misclassified instances could be deceivingly high. For cases where a false negative incurs a greater penalty than a false positive, a learner's bias in favor of the majority class may have adverse consequences [19, 20]. As an example, in a national hospital's database of patients with migraine, a very small number will most likely test positive for brain cancer, i.e., minority class, while most are expected to test negative, i.e., majority class. In this situation, a false negative means that a patient with brain cancer has been misclassified as not having the disease, which is a very grave error.

Our contribution shows how changes in data distribution over time affect predictability with regard to the maintenance of machine learning models. To achieve this, we utilize five learners, five class ratios obtained by RUS, the *Area Under the Receiver Operating Characteristic Curve* (AUC) metric, and three Medicare datasets. Upon evaluation, the top models selected are as follows: *Logistic Regression* with the 2015 year-grouping at a 99:1 class ratio (Part D); *Random Forest* with the 2014-2015 year-grouping at a 75:25 class ratio (DMEPOS); and *Logistic Regression* with the full 2015 year-grouping (Combined). Empirical results indicate that the largest training dataset, i.e., year-grouping 2013–2015, was not among the selected choice of models, thus suggesting that the 2013 data may be dated and not beneficial for Medicare fraud detection. Since the top choice of model is different for the Part D, DMEPOS, and Combined datasets, we postulate that each of the three datasets may be a unique subdomain under the broader domain of Medicare fraud detection. To the best of our knowledge, we are the first to investigate, through the use of several big datasets, the effect of time on predictive model maintenance.

The remainder of this paper is organized as follows: "Related work" section covers related literature investigating the relationship between time and predictive model maintenance; "Datasets" section describes the Medicare datasets, along with our data processing approach; "Learners" section provides information on the learners and their configuration settings; "Methodologies" section describes the different aspects of the methodology used to develop and implement our approach; "Results and discussion" section presents and discusses our empirical results; and "Conclusion" section concludes our paper with a summary of the research work and suggestions for related future work.

Related work

There are various approaches for addressing data distribution changes with time [2]. However, apart from our recently published conference paper [21], we could not find other research works that investigate the relationship between time and predictive model maintenance for big data.

In [22], Raza et al. proposed a solution that detects dataset shift with an *Exponentially Weighted Moving Average* (EWMA) control chart. The chart is an established statistical method for identifying minor shifts in time-series data [23], and it joins past and current data to enable rapid detection of these shifts. The researchers evaluated model performance using both real-world and synthetic datasets. Through their model, Raza et al. attempted to detect the shift-point in a data stream. A shift-point is a marked change in one or more slopes of a linear time-series model [24]. The work in [22] is limited due to the EWMA chart assigning a time-weighted constant to past and current observations, where the inclusion of an incorrect constant could lead to shift-point misidentification.

Ikonomovska et al., in [25], recognized that data streams is a promising research area, and they developed a regression tree algorithm for identifying variations in data distributions. Starting with an empty leaf node, the algorithm sequentially reads instances and determines the best split for each feature. Features are then ranked, with the most favorable attribute split if a specific threshold is reached. Each data stream instance that arrives triggers a change detection check, and on detection of a change, the tree structure is updated. We note that if the tree model becomes too large, the model becomes complex and interpretability suffers.

With *Multivariate Relevance Vector Machines* [26], Torres et al. [27] examined direct and indirect approaches for forecasting daily evapotranspiration. The estimation of evaporation is important for water management and irrigation scheduling. Utilizing the *Multilayer Perceptron* (MLP) [28] learner as a benchmark, the researchers calculated potential crop evapotranspiration and compared crop value results in their study location. Their results showed that it is possible to accurately forecast up to four days of potential crop evapotranspiration, and that the indirect approach outperformed both the direct approach and MLP learner. In other words, after four days the phenomenon of concept drift adversely affected model performance. The study is limited by the use of only one learner as a benchmark.

Using an Adaboost-*Support Vector Machines* (SVMs) ensemble, Sun et al. [29] implemented time-based weighting on data batches to predict dynamic financial distress. This distress is associated with conditions such as bankruptcy and debt default [30]. The researchers designed two different algorithms for predicting financial distress. The first merges the outputs of a time-based and error-based decision expert system, and the second applies a time-based weight updating function during iterations of Adaboost. As noted previously, the use of time weighting could impact the effectiveness of study results.

Finally, in our recent conference paper [21], we examined the effect of time on the maintenance of a predictive model to detect Medicare Part B billing fraud. Training datasets were built from year-groupings of 2015, 2014–2015, 2013–2015, and 2012–2015, while the test datasets were built from 2016 data. Our study incorporated five class ratios obtained by RUS, and five learners. Using the AUC performance metric, we showed that the *Logistic Regression* learner produces the highest overall value for the year-grouping of 2013–2015, with a majority-to-minority ratio of 90:10. Furthermore, we concluded that a sampled dataset should be selected over the full dataset and that the largest training dataset, i.e., 2012–2015, does not always yield the best results. The work in [21] is limited to the Medicare Part B dataset, and therefore, in our current paper we remove this limitation by performing experimentation on Part D, DMEPOS, and Combined Medicare datasets.

Throughout our search for related works, we observed that existing literature on the dynamism of data distribution is mainly centered around data streams, i.e., real-time data. In our current paper, however, constructed models are based on static datasets, meaning that our predictive models have been trained on static data that was collected and processed in an offline mode. On the other hand, online processing is a requirement for data streams because this data arrives in real time and may overburden the computer system. For these online cases, predictive models are retrained with recent data batches or incrementally trained [2]. Although investigating the dynamic nature of data distribution for static databases and data streams are both equally important, it is easier to observe the distribution variations with real-time data. On account of this, static databases are frequently omitted from such studies.

Datasets

The *Centers for Medicare and Medicaid Services* (CMS) datasets used in this work (Part B, Part D, DMEPOS, and Combined) are discussed in this section, along with the data processing methodology and also, the LEIE dataset that provides the fraud labels. Our training and test datasets are derived from these original CMS and LEIE datasets. CMS records all claims information after payments are disbursed [31–33], and therefore, we consider the Medicare data to be cleansed and accurate. We note that *National Provider Identifier* (NPI) [34] is utilized for aggregation and identification, but not during the data mining stage. Furthermore, a year variable was added for each dataset.

Part B

The Part B dataset contains claims information for each procedure a physician performs in a specific year [35]. Physicians are identified by their unique NPI, and procedures are assigned their respective *Healthcare Common Procedure Coding System* (HCPCS) codes [36]. The number of procedures performed, average payments and charges, and medical specialty (referred to as provider type) are also covered by Medicare Part B. CMS has aggregated Part B data by NPI, HCPCS code, and the place of service (facility (F) such as a hospital, or non-facility (O) such as an office). Every dataset row includes an NPI, provider type, one HCPCS code matched to place of service along with related information, and other static features such as gender. For each physician, each dataset row represents a unique combination of NPI, provider type, HCPCS code, and place of service. Note that the Part B dataset is not the focus of our paper. However, Part B is a component of the Combined, a dataset integral to our current work.

Part D

The Part D dataset contains information on prescription drugs provided under the Medicare Part D Prescription Drug Program in a specific year [37]. Physicians are identified by their unique NPI and drugs are labeled according to their brand and generic name. Other information contained in the dataset includes average payments and charges, variables describing the drug quantity prescribed, and medical specialty. CMS has aggregated Part D data by NPI and the drug name. Every dataset row includes an NPI, provider type, drug name along with related information, and other static features such as gender. For each physician, each dataset row represents a unique combination of NPI, provider type, and drug name. Aggregated records, obtained from fewer than 11 claims, are omitted from the Part D data. This is done to safeguard the privacy of Medicare beneficiaries.

DMEPOS

The DMEPOS dataset contains information on Medical Equipment, Prosthetics, Orthotics and Supplies that physicians referred their patients to either buy or rent from a supplier in a specific year [38]. This dataset is derived from claims that suppliers have submitted to Medicare. The role of the physician in this case is to refer the patient to the supplier. Physicians are identified by their unique NPI [34], and products are assigned their HCPCS code. Other claims information includes the number of services/products rented or sold, average payments and charges, and medical specialty. CMS has aggregated DMEPOS data by NPI, HCPCS code, and supplier rental indicator obtained from DMEPOS supplier claims. Every dataset row includes an NPI, provider type, one HCPCS code matched to place of service along with related information, and other static features such as gender. For each physician, each dataset row represents a unique combination of NPI, provider type, HCPCS code and rental status.

LEIE

A dataset of physicians who committed fraud is necessary to accurately assess fraud detection performance in the real world. For this reason, we utilized the LEIE [39], which provides information such as reason for exclusion and date of exclusion. The LEIE was established by the *Office of Inspector General* (OIG) [40], which has a mandate to exclude individuals and entities from federally funded healthcare programs. We note, however, that the LEIE dataset contains NPI values for only a fraction of fraudulent physicians and entities in the US. Nationally, approximately 21% of convicted fraudulent providers have not been suspended from medical practice, and about 38% of those convicted continue to practice medicine [41].

The LEIE does not provide specific information relating to drugs, equipment, or procedures involving fraudulent activities. There are several types of exclusions that are described by various rule numbers, and we selected only rules indicating fraud was committed, as shown in Table 1 [42].

Table 1 Selected LEIE rules

Rule number	Description	Exclusion period
1128(a)1	Conviction of program-related crimes	5 years
1128(a)2	Conviction due to patient abuse or neglect	5 years
1128(a)3	Felony conviction due to healthcare fraud	5 years
1128(b)4	License revocation or suspension	5 years
1128(b)7	Fraud, kickbacks and other prohibited activities	5 years
1128(c)(3)(g)(i)	Conviction of two mandatory exclusion offenses	10 years
1128(c)(3)(g)(ii)	Conviction of 3 mandatory exclusion offenses	Indefinite

Table 2 Selected features from datasets

Dataset	Feature	Description
Part B[a]	npi	Unique provider identification number
	provider_type	Medical provider's specialty (or practice)
	nppes_provider_gender	Provider's gender
	line_srvc_cnt	Number of procedures/services the provider performed
	bene_unique_cnt	Number of distinct Medicare beneficiaries receiving the service
	bene_day_srvc_cnt	Number of distinct Medicare beneficiaries / per day service performed
	average_submitted_chrg_amt	Average of the charges that the provider submitted for the service
	average_medicare_payment_amt	Average payment made to a provider per claim for the service performed
Part D	npi	Unique provider identification number
	specialty_description	Medical provider's specialty (or practice)
	bene_count	Number of distinct Medicare beneficiaries receiving the drug
	total_claim_count	Number of drugs the provider administered
	total_30_day_fill_count	Number of standardized 30-day fills
	total_day_supply	Number of day's supply
	total_drug_cost	Cost paid for all associated claims
DMEPOS	referring_npi	Unique provider identification number
	referring_provider_type	Medical provider's specialty (or practice)
	referring_provider_gender	Provider's gender
	number_of_suppliers	Number of suppliers used by provider
	number_of_supplier_beneficiaries	Number of beneficiaries associated by the supplier
	number_of_supplier_claims	Number of claims submitted by a supplier due to an order by a referring order
	number_of_supplier_services	Number of services/products rendered by a supplier
	avg_supplier_submitted_charge	Average payment submitted by a supplier
	avg_supplier_medicare_pmt_amt	Average payment awarded to suppliers

[a] Part B alone is not used to train and test models in this paper

Data processing

When this study was conducted, Part B was available for 2012 through 2016, while Part D and DMEPOS were available for 2013 through 2016. We selected specific attributes among the three datasets in order to provide a solid foundation for our analyses. Also, for consistency purposes, the 2012 data of Part B was removed. For Part B, Part D, and DMEPOS, we selected eight, seven and nine features, respectively.

Excluded features contained no information on drugs provided, claims, or referrals, but instead provided provider-related information, such as location and name, as well as redundant variables. Table 2 shows the features that we selected from each original dataset [9].

All three original datasets are at the procedure level, which means they were aggregated by NPI and HCPCS codes. To conform to our methodology of mapping fraud labels with LEIE, each dataset was aggregated to the provider-level, a rearrangement that groups all information over each NPI (and other particular attributes) [43]. For each numeric value per year, we replace the variable in each dataset with the aggregated mean, median, sum, standard deviation, minimum and maximum values, creating six new attributes for each original numeric attribute.

Combined dataset

The Combined dataset entails a join operation on NPI, provider type, and year for Part B, Part D, and DMEPOS, after individual processing of these datasets [43]. As Part D contains no gender variable, this feature was not included [43]. Note that the combining of these datasets limits us to physicians who have participated in all three parts of Medicare. However, the Combined has more numerous and inclusive features than the other three Medicare datasets.

Fraud labeling

For our processed Medicare datasets, we obtain fraud labels from the LEIE dataset [43]. Only physicians within the LEIE are considered fraudulent for the purpose of this study. This dataset is joined to the Medicare datasets by NPI, and physicians practicing within a year prior to their exclusion end year are labeled fraudulent. Table 3 shows the distribution of fraud to non-fraud within the full datasets [9], which are highly or severely

Table 3 Full datasets

Dataset	Year	Total instances	Fraud instances	Fraud %
Part B[a]	2013	915,909	403	0.044
	2014	950,000	285	0.030
	2015	972,222	175	0.018
	2016	990,000	99	0.010
Part D	2013	673,913	465	0.069
	2014	700,000	329	0.047
	2015	722,580	224	0.031
	2016	750,000	135	0.018
DMEPOS	2013	293,636	323	0.110
	2014	283,823	193	0.068
	2015	290,243	119	0.041
	2016	288,461	75	0.026
Combined	2013	254,444	229	0.090
	2014	252,459	154	0.061
	2015	257,142	90	0.035
	2016	261,904	55	0.021

[a] Part B alone is not used to train and test models in this paper

imbalanced. Note that year-groupings can correspond to single years, as in the case of 2015, or a combination of years, as in the case of 2013-2015. Part B, which is a component of the Combined but not the focus of this paper, is shown in the table for informational purposes only.

One-hot encoding

One-hot encoding is used in our model construction to transform categorical features into numerical ones [43]. For instance, one-hot encoding of gender generates extra features equal to the number of options (male and female). If the physician is male, the new male feature would be assigned a 1 and the female feature a 0, and vice-versa if the physician is female. Both male and female features could be assigned a 0 in cases where the original gender feature is not provided.

Learners

Our work uses five popular learners (*k-NearestNeighbor* (k-NN), C4.5 decision tree, *Random Forest* (RF), LR, *Support Vector Machine* (SVM)), all of which are available within *Waikato Environment for Knowledge Analysis* (WEKA), an open source collection of machine learning algorithms. These classifiers were chosen for their good coverage of several *Machine Learning* (ML) model families. Performance-wise, the five classifiers are regarded favorably, and they incorporate both ensemble and non-ensemble algorithms [44, 45]. In this section, we describe each model and note configuration and hyperparameter changes that differ from the default settings in WEKA.

The *k*-NN learner [46], also called IBk (Instance Based Learner with parameter *k*) in WEKA, specifies the number of nearest neighbors to use for classification and implements distance-based comparisons among instances. The performance of KNN relies on the distance measure, with Euclidean distance being the typical choice. We assigned a value of 5 to *k* (5-NN), and set the 'distanceWeighting' parameter as 'Weight by 1/distance' in order to use inverse distance weighting for determining class membership [47].

C4.5 decision tree [48] uses a divide-and-conquer approach to split the data at each node based on the feature with the most information. Node attributes are automatically chosen by maximizing information gain and minimizing entropy. Entropy is a measure of the uncertainty of attributes, with information gain being the means to find the most informative attribute. Features that are most valuable are located near the root node, and the leaf nodes contain the classification results. The J48 decision tree is the standard implementation within WEKA. We set the J48 parameters to 'Laplace Smoothing' and 'no pruning', which can improve results for imbalanced data [47].

RF [49] is an ensemble technique for assembling multiple, unpruned decision trees into a forest. Class membership is calculated by combining the results of the individual trees, usually by majority voting. Through sampling with replacement, RF produces random datasets to build each decision tree. Node features are automatically chosen based on entropy and information gain. In addition, RF uses feature subspace selection to randomly assign *i* features for each tree. Since RF is a random ensemble technique, data is not likely to be overfitted. With preliminary analysis indicating no difference between 100 and 500 trees, our RF learners were constructed with only 100 trees [47].

Logistic Regression [50] utilizes a sigmoid function to produce values from [0,1], which translates into class probabilities. A sigmoid function is a special case of the logistic function. LR, unlike linear regression, predicts class membership by means of a separate hypothesis class. We did not change the default setting in WEKA for the 'ridge' parameter, which is the penalized maximum likelihood estimation with a quadratic penalty function (also called L2 regularization) [47].

The *Support Vector Machine* learner [51] assumes that class instances are linearly separable and uses hyperplanes to separate them. The hyperplane maximizes the distance between the two classes. SVM uses regularization to prevent overfitting via the complexity parameter 'c'. In WEKA, we set the complexity parameter 'c' to 5.0. The 'buildLogisticModels' parameter, which allows probability estimates to be returned, was set to true [47].

Methodologies

Performance metric

Accuracy is often obtained from a simple 0.50 threshold that is incorporated into a formula for predicting one out of the two binary classes. For most real-world situations, however, the two classes are imbalanced, leading to a majority and minority class grouping. The *Confusion Matrix* (CM) for a binary classification problem is depicted in Table 4 [20], where Positive, the class of interest, is the minority class and Negative is the majority class.

- *True Positive* (TP) are positive instances correctly identified as positive.
- *True Negative* (TN) are negative instances correctly identified as negative.
- *False Positive* (FP), also known as Type I error, are negative instances incorrectly identified as positive.
- *False Negative* (FN), also known as Type II error, are positive instances incorrectly identified as negative.

Based on these four fundamental CM metrics, other performance metrics that consider the rates between the positive and the negative class are derived as follows:

- *True Positive Rate* ($\mathrm{TP}_{\mathrm{rate}}$), also known as Recall or Sensitivity, is equal to TP/(TP + FN).
- *True Negative Rate* ($\mathrm{TN}_{\mathrm{rate}}$), also known as Specificity, is equal to TN/(TN + FP).

Table 4 Confusion Matrix

Actual class	**Predicted class**	
	Positive	**Negative**
Positive	True Positive (TP)	False Negative (FN) (Type II error)
Negative	False Positive (FP) (Type I error)	True Negative (TN)

- *False Positive Rate* (FP_{rate}), also known as false alarm rate, is equal to FP/(FP + TN), which usually refers to the expectancy of the false positive ratio.
- *Positive Predictive Value* (PPV), also known as Precision, is equal to TP/(TP + FP).

The AUC metric calculates the area under the *Receiver Operating Characteristic* (ROC) curve, which graphically shows TP_{rate} versus FP_{rate} for various classification cut-offs. AUC represents the behavior of a classifier across all thresholds of the ROC curve and is a popular metric that mitigates the negative effects of class imbalance [47]. A model whose predictions are 100% correct has an AUC of 1, while a model whose predictions are 100% incorrect has an AUC of 0.

Model evaluation

A popular evaluation method in ML is train-test, in which one dataset trains the model while a separate dataset tests the model, with all instances in the test dataset completely new [9]. The train-test method determines whether, based on past occurrences, a model can accurately predict new occurrences. For our study, the train-test method indicates whether, based on previous information (year < 2016), physicians can be classified as fraudulent or non-fraudulent given new information (year = 2016).

Machine learning framework

All learners in our study were implemented within WEKA [16], an open source framework of machine learning techniques issued under the GNU General Public License. Written in Java, this framework is used for various types of machine learning tasks, such as data preparation, classification, and regression. The graphical user interfaces of WEKA contribute to its ease of use, with the software being widely used by ML researchers, industrial scientists, and students.

Random Undersampling

RUS is beneficial for imbalanced big data, as removing instances decreases computational burden and build time [52, 53]. When applying RUS, the aim is to strike a balance between discarding the maximum number of majority instances while incurring the least information loss. For our research, we selected the following majority-to minority class ratios: 50:50, 65:35, 75:25, 90:10, and 99:1. These ratios were chosen because they collectively provide good distribution coverage, ranging from the balanced ratio of 50:50 to the highly imbalanced ratio of 99:1. In addition, we included the full datasets as the baseline, where RUS is not applied. For the train-test method, only the training datasets were sampled. The test datasets were only used for model evaluation, and therefore were not sampled.

Addressing randomness

Since a sample of instances is often relatively small compared to its respective original dataset, the randomization process in RUS may result in information loss, thus impacting classification performance [54]. This also means that the classification outcome could differ each time RUS is carried out, creating splits that may be deemed favorable, fair, or unlucky to the learner. Splits viewed as favorable may retain very good or clean instances that improve learner performance, but could potentially overfit the model. On the other hand, unlucky splits may retain noisy instances that weaken classification performance.

It is worth noting that some ML algorithms, such as RF, have an inherent randomness within their implementation. Furthermore, the random shuffling of instances performed before the start of each training process may cause other algorithms, such as LR, to produce different results if the order of instances is altered.

The use of repetitive methods is a proven technique for reducing the potential negative effects of randomness [55]. To address randomness during our sampling and model building stages, we performed ten repetitions per built model and selected the average of each set of repetitions.

Experiment design

This subsection highlights the main points of "Methodologies" section. To mitigate the adverse effect of high class imbalance in the full training datasets, RUS was applied. Using RUS, we obtained the following class ratios: 50:50, 65:35, 75:25, 90:10, and 99:1. The AUC learner was selected to evaluate classifier performance as it helps correct the distortion of results due to class imbalance [47]. Within the WEKA framework, model prediction was evaluated against the 2016 test set. The evaluation process was repeated ten times per training dataset, with average results reported.

Results and discussion

Table 5 shows the mean AUC values for the five learners for Part D, DMEPOS, and Combined datasets, with values ranked in descending order for each dataset. Individual rows in each table are distinguished by their specific combination of sampled class ratio and year-grouping. The term "None_Full" indicates that the full dataset, with no sampling, was used as training data.

For Part D, the highest value (0.8167) corresponds to LR with the 2015 year-grouping at a 99:1 class ratio. The lowest value (0.7567) is associated with C4.5 decision

Table 5 AUC summary

Dataset	Distribution ratio	Training set	Test set	AUC
Part D	LR_RUS_99:1	2015	2016	0.8167
	SVM_RUS_65:35	2013-2015	2016	0.7936
	RF100_RUS_75:25	2013-2015	2016	0.7834
	5NN_RUS_65:35	2014-2015	2016	0.7570
	C4.5_RUS_99:1	2014-2015	2016	0.7567
DMEPOS	RF100_RUS_75:25	2014-2015	2016	0.7872
	C4.5_None_Full	2014-2015	2016	0.7845
	LR_None_Full	2014-2015	2016	0.7834
	SVM_RUS_50:50	2014-2015	2016	0.7773
	5NN_RUS_65:35	2015	2016	0.7721
Combined	LR_None_Full	2015	2016	0.8738
	SVM_RUS_75:25	2013-2015	2016	0.8691
	RF100_RUS_90:10	2013-2015	2016	0.8680
	5NN_RUS_75:25	2015	2016	0.8199
	C4.5_RUS_65:35	2014-2015	2016	0.7650

tree for the 2014–2015 year-grouping at a 99:1 class ratio. With regard to DMEPOS, the highest value (0.7872) equates to RF with the 2014–2015 year-grouping at a 75:25 class ratio. The lowest value (0.7721) comes from (5-NN) with the 2015 year-grouping at a 65:35 class ratio. Finally, for Combined, the highest value (0.8738) was obtained from LR with the full 2015 year-grouping. The lowest value (0.7650) is associated with C4.5 decision tree for the 2014–2015 year-grouping at a 65:35 class ratio. As can be observed in Table 5, DMEPOS has the smallest spread of mean AUC values, while Combined has the largest.

Based on the discussion in the previous paragraph, the top choices are as follows: LR with the 2015 year-grouping at a 99:1 class ratio (Part D); RF with the 2014–2015 year-grouping at a 75:25 class ratio (DMEPOS); LR with the full 2015 year-grouping (Combined). In order to confirm these combinations as our top choices, further insight is needed on the performance of LR and RF. Therefore, Figs. 2 through 7 have been included.

Box plots are shown in Figs. 2, 3, and 4, which represent LR for Part D, RF for DMEPOS, and LR for Combined, respectively. A box plot depicts the median (50th percentile) as a thick line, two hinges (25th and 75th percentiles), two whiskers, and outlying points. With regard to Figure 2, at the 99:1 ratio, the 2015 box does not overlap with other year-groupings for LR. This indicates that the difference between year-grouping 2015 and the other year-groupings is significant. An analysis of Fig. 3 indicates that at the 75:25 ratio, the 2014–2015 box does not overlap with other year-groupings for RF, which translates into a significant difference between year-grouping 2014–2015 and the rest of the year-groupings. In Fig. 4, it is obvious there is no overlap with the short, vertical line representing 2015 and the lines for the other year-groupings, indicating that the difference between year-grouping 2015 and the other year-groupings is significant.

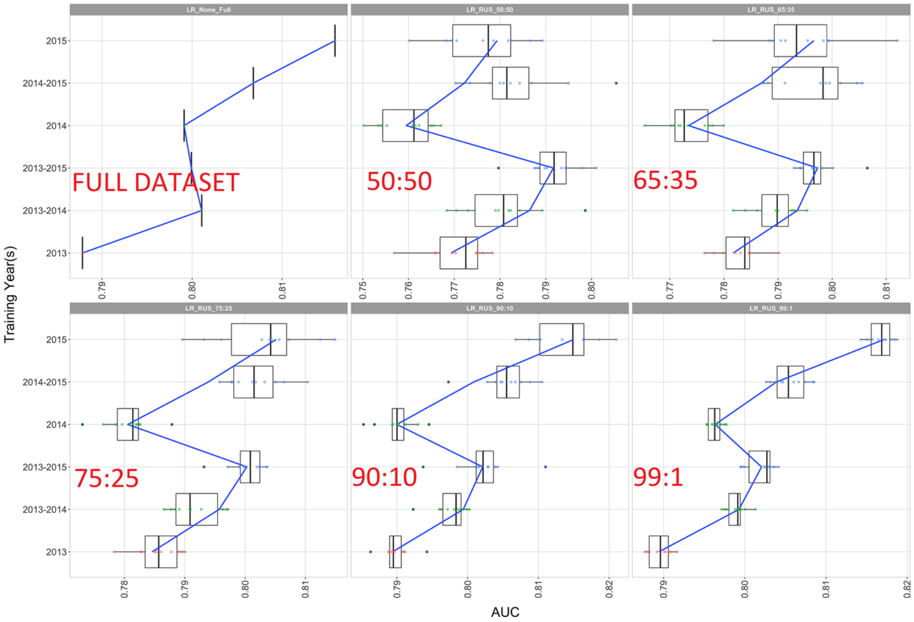

Fig. 2 Logistic Regression box plots (Part D)

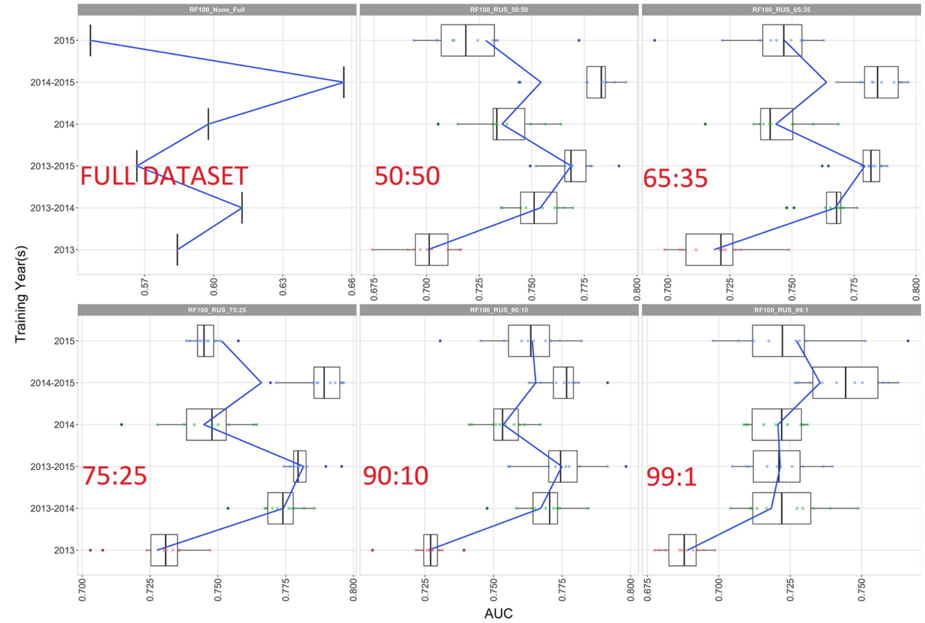

Fig. 3 Random Forest box plots (DMEPOS)

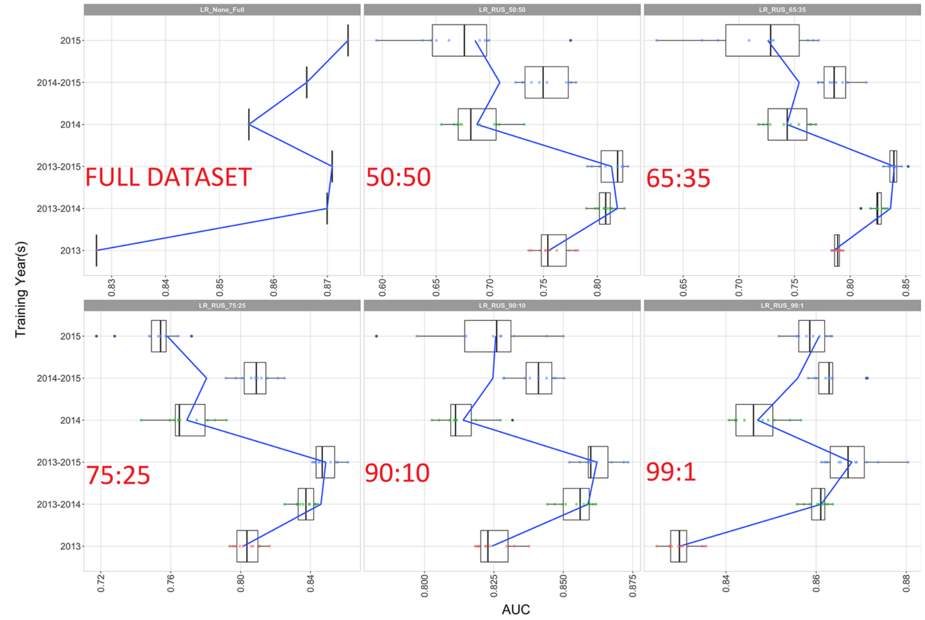

Fig. 4 Logistic Regression box plots (combined)

This section is a report on our research methodologies, including reasons for choosing them. We discuss the performance metric, model evaluation, machine learning framework, Random Undersampling, addressing randomness, and experiment design.

Figures 5, 6, and 7 show Tukey's *Honestly Significant Difference* (HSD) plots representing LR for Part D, RF for DMEPOS, and LR for Combined, respectively. Each vertical bar represents the AUC score of a group (year-grouping or distribution ratio) for

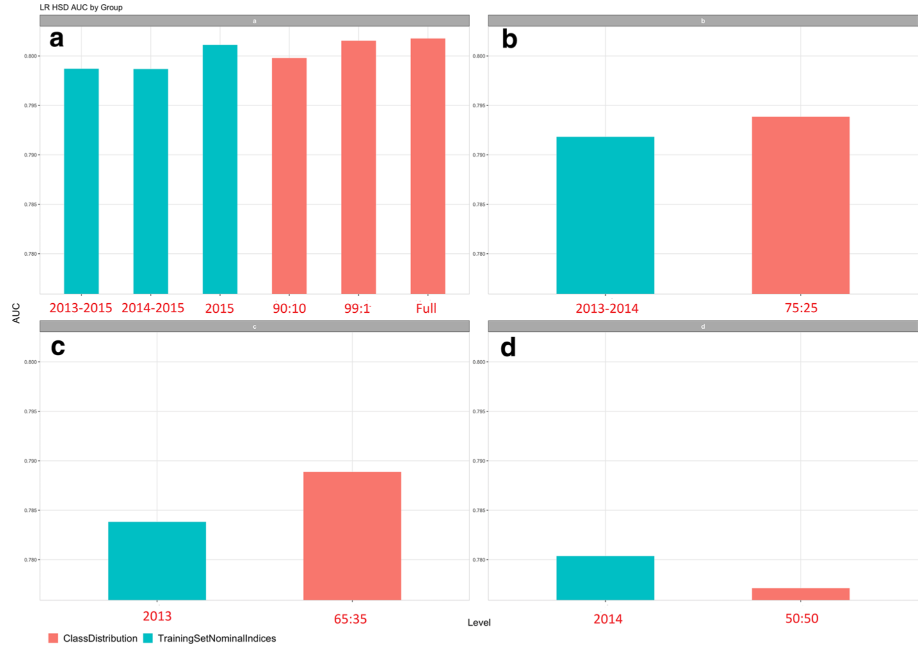

Fig. 5 Logistic Regression HSD plots (Part D)

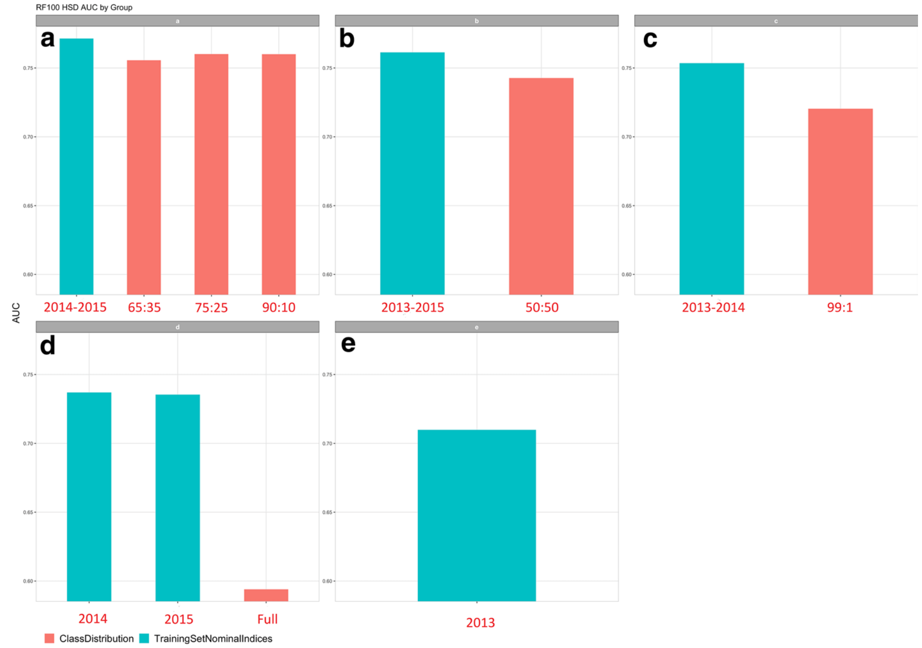

Fig. 6 Random Forest HSD plots (DMEPOS)

a specific learner. A Tukey's HSD [56] test determines the group factors that are significant. For our experiments, we use a 5% significance level. Letter groups assigned by the test denote similarity or significant differences in results within each group or factor, with an 'a' representing the top group.

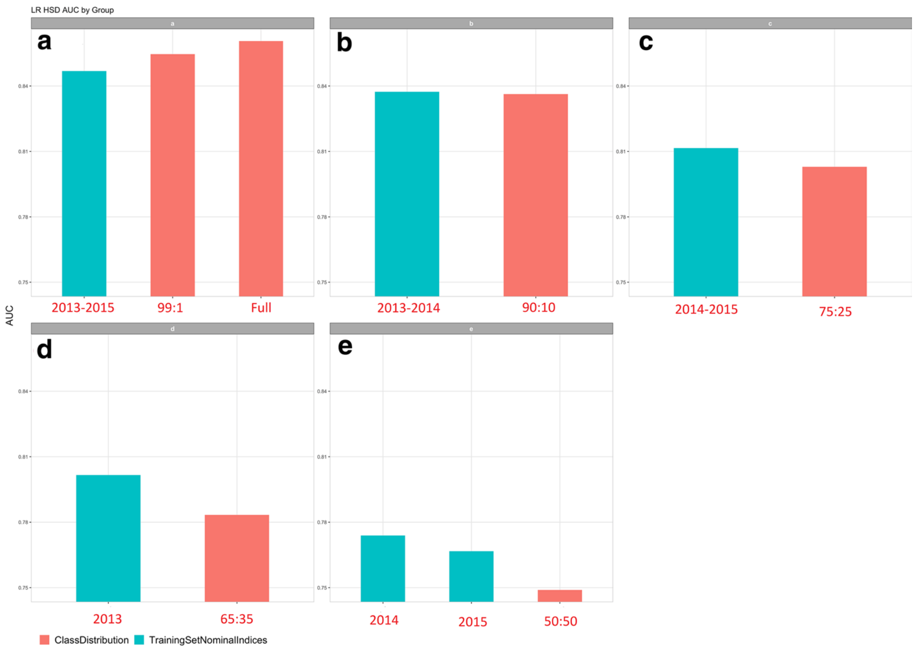

Fig. 7 Logistic Regression HSD plots (combined)

In Fig. 5, the plots reveal that year-groupings 2013–2015, 2014–2015, and 2015, and the full datasets along with class ratios 90:10 and 99:1 have a group 'a' ranking for LR. With respect to Fig. 6, the plots show that year-grouping 2014–2015 and class ratios 65:35, 75:25, and 90:10 have a group 'a' ranking for RF. Lastly, for Fig. 7, the plots indicate that year-grouping 2013-2015 and the full datasets along with the 99:1 class ratio have a group 'a' ranking for LR.

Therefore, we conclude from Figs. 2 through 7 that our top choices should be selected: LR with the 2015 year-grouping at a 99:1 class ratio (Part D); RF with the 2014-2015 year-grouping at a 75:25 class ratio (DMEPOS); and LR with the full 2015 year-grouping (Combined). It is important to note that the largest training dataset (2013–2015) did not feature among our top choices. We attribute this outcome to the likelihood that the 2013 data is outdated. In addition, it is obvious that the top choice of model for Part D, DMEPOS, and Combined is not the same. Although Part D and Combined share the same LR learner and 2015 year-grouping for their top choice, experimental results show that RUS is needed to produce the best results for the former, while the full dataset yields the best results for the latter. This disparity in top choice hints that for fraud detection purposes, Part D, DMEPOS, and Combined datasets are sub-domains within the main domain of Medicare fraud detection. Each sub-domain would have a distinct data distribution, and thus require a model and RUS distribution that are also unique.

Conclusion

The regular updating of machine learning models is necessary because their original data distributions tend to change over time. These temporal changes are often detrimental to predictive effectiveness. In this paper, we analyze the impact of incorporating training data from several year-groupings on an existing predictive model that detects fraud in Medicare datasets. Our training datasets are constructed from year-groupings of 2013, 2014, 2015, 2013–2014, 2014–2015, and 2013–2015, while our test datasets were built from 2016 data. We use five class ratios obtained by *Random Undersampling*, five popular learners, and the *Area Under the Receiver Operating Characteristic Curve* performance metric.

Based on our results, we determined that the following models should be used in order to yield the top results: Part D-LR with the 2015 year-grouping at a 99:1 class ratio; DMEPOS-RF with the 2014-2015 year-grouping at a 75:25 class ratio; and Combined - LR with the full 2015 year-grouping. The reader should appreciate the fact that the largest year-grouping of training data (2013–2015) did not produce the highest AUC values, which signals that the 2013 data may be outdated. In addition, we note that because the top choice for predictive model is different for Part D, DMEPOS, and Combined, this suggests that these three datasets, for the purposes of Medicare fraud detection, may be sub-domains.

Future work will examine the effect of using learners, class ratios and performance metrics that are different from those utilized in this study, and also investigate the impact of sourcing big data from different application domains.

Abbreviations

RUS : Random Undersampling; **ML** : Machine Learning; **LEIE** : List of Excluded Individuals/Entities; **RF** : Random Forest; **LR** : Logistic Regression; **SVMs** : Support Vector Machines; **SVM** : Support Vector Machine; k-**NN** : k-Nearest Neighbor; **5-NN** : 5-Nearest Neighbors; **WEKA** : Waikato Environment for Knowledge Analysis; **CM** : Confusion Matrix; **CMS** : Centers for Medicare and Medicaid Services; **NPI** : National Provider Identifier; **HCPCS** : Healthcare Common Procedure Coding System; **LEIE** : List of Excluded Individuals/Entities; **OIG** : Office of Inspector General; **DMEPOS** : Durable Medical Equipment, Prosthetics, Orthotics and Supplies; **MVRVM** : Multivariate Relevance Vector Machines; **EWMA** : Exponentially Weighted Moving Average; **MLP** : Multilayer Perceptron; **ROC** : Receiver Operating Characteristic; **AUC** : Area Under the Receiver Operating Characteristic Curve; $\mathbf{TP_{rate}}$: True Positive Rate; $\mathbf{TN_{rate}}$: True Negative Rate; $\mathbf{FP_{rate}}$: False Positive Rate; **TP** : True Positive; **TN** : True Negative; **FP** : False Positive; **FN** : False Negative; **PPV** : Positive Predictive Value; **HSD** : Honestly Significant Difference; **NSF** : National Science Foundation.

Acknowledgements

We would like to thank the reviewers in the Data Mining and Machine Learning Laboratory at Florida Atlantic University. Additionally, we acknowledge partial support by the *National Science Foundation* (NSF) (CNS-1427536). Opinions, findings, conclusions, or recommendations in this paper are the authors' and do not reflect the views of the NSF.

Authors' contributions

JLL and RAB conceived and designed the research, performed the implementation and experimentation, and performed the evaluation and validation. RAB prepared the Medicare datasets. JLL performed the primary literature review for this work and drafted the manuscript. All authors provided feedback to JLL and helped shape the research. TMK introduced this topic to JLL, and helped to complete and finalize this work. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA. [2] Ohio Northern University, 525 South Main Street, Ada, OH 45810, USA.

Received: 12 February 2020 Accepted: 25 May 2020
Published online: 05 June 2020

References

1. Žliobaitė I, Pechenizkiy M, Gama J. An overview of concept drift applications. In: Big Data Analysis: New Algorithms for a New Society. Switzerland: Springer; 2016. 91–114
2. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Comput Surv. 2014;46(4):44.
3. Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts. Mach Learn. 1996;23(1):69–101.
4. Tsymbal A. The problem of concept drift: definitions and related work. Comput Sci Dep Trinity College Dublin. 2004;106(2):58.
5. Moreno-Torres JG, Raeder T, Alaiz-RodríGuez R, Chawla NV, Herrera F. A unifying view on dataset shift in classification. Pattern Recognit. 2012;45(1):521–30.
6. Turhan B. On the dataset shift problem in software engineering prediction models. Empir Softw Eng. 2012;17(1–2):62–74.
7. Vaze J, Post D, Chiew F, Perraud J-M, Viney N, Teng J. Climate non-stationarity-validity of calibrated rainfall-runoff models for use in climate change studies. J Hydrol. 2010;394(3–4):447–57.
8. Chilakapati A. Concept drift and model decay in machine learning. http://xplordat.com/2019/04/25/concept-drift-and-model-decay-in-machine-learning/ 2019.
9. Herland M, Bauder RA, Khoshgoftaar TM. The effects of class rarity on the evaluation of supervised healthcare fraud detection models. J Big Data. 2019;6(1):21.
10. Katal A, Wazid M, Goudar R. Big data: issues, challenges, tools and good practices. In: 2013 sixth international conference on contemporary computing (IC3). New York: IEEE; 2013. 404–409.
11. Huang G-B, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B. 2011;42(2):513–29.
12. Japkowicz N, Stephen S. The class imbalance problem: a systematic study. Intel Data Anal. 2002;6(5):429–49.
13. Leevy JL, Khoshgoftaar TM, Bauder RA, Seliya N. A survey on addressing high-class imbalance in big data. J Big Data. 2018;5(1):42.
14. Maurya A. Bayesian optimization for predicting rare internal failures in manufacturing processes. In: 2016 IEEE international conference on big data (big data). New York: IEEE; 2016. 2036–2045.
15. He H, Garcia EA. Learning from imbalanced data. IEEE Trans knowl Data Eng. 2009;21(9):1263–84.
16. Witten IH, Frank E, Hall MA, Pal CJ. Data mining: practical machine learning tools and techniques. Burlington: Morgan Kaufmann; 2016.
17. Olden JD, Lawler JJ, Poff NL. Machine learning methods without tears: a primer for ecologists. Q Rev Biol. 2008;83(2):171–93.
18. Galindo J, Tamayo P. Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. Comput Econ. 2000;15(1):107–43.
19. Akay MF. Support vector machines combined with feature selection for breast cancer diagnosis. Expert Syst Appl. 2009;36(2):3240–7.
20. Seliya N, Khoshgoftaar TM, Van Hulse J. A study on the relationships of classifier performance metrics. In: 2009 21st IEEE international conference on tools with artificial intelligence. New York: IEEE; 2009. 59–66.
21. Leevy JL, Khoshgoftaar TM, Bauder RA, Seliya N. The effect of time on the maintenance of a predictive model. In: 2019 18th IEEE international conference on machine learning and applications (ICMLA). New York: IEEE; 2019
22. Raza H, Prasad G, Li Y. Dataset shift detection in non-stationary environments using ewma charts. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics. New York: IEEE; 2013. 3151–3156.
23. Roberts S. Control chart tests based on geometric moving averages. Technometrics. 1959;1(3):239–50.
24. Farley JU, Hinich M, McGuire TW. Some comparisons of tests for a shift in the slopes of a multivariate linear time series model. J Econ. 1975;3(3):297–318.
25. Ikonomovska E, Gama J, Džeroski S. Learning model trees from evolving data streams. Data Mining Knowl Discov. 2011;23(1):128–68.
26. Thayananthan A, Navaratnam R, Stenger B, Torr PH, Cipolla R. Multivariate relevance vector machines for tracking. In: European conference on computer vision. Berlin: Springer; 2006. 124–138
27. Torres AF, Walker WR, McKee M. Forecasting daily potential evapotranspiration using machine learning and limited climatic data. Agric Water Manag. 2011;98(4):553–62.
28. Gardner MW, Dorling S. Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences. Atmos Environ. 1998;32(14–15):2627–36.
29. Sun J, Fujita H, Chen P, Li H. Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble. Knowl Based Syst. 2017;120:4–14.
30. Sun J, He K-Y, Li H. Sffs-pc-nn optimized by genetic algorithm for dynamic prediction of financial distress with longitudinal data streams. Knowl Based Syst. 2011;24(7):1013–23.
31. Of Enterprise Data, C.O., Analytics: Medicare Fee-For-Service Provider Utilization & Payment Data Physician and Other Supplier. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Medicare-Physician-and-Other-Supplier-PUF-Methodology.pdf
32. Of Enterprise Data, C.O., Analytics: Medicare Fee-For Service Provider Utilization & Payment Data Part D prescriber public use file: a methodological overview. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Prescriber_Methods.pdf
33. Of Enterprise Data, C.O., Analytics: Medicare Fee-For-Service Provider Utilization & Payment Data Referring durable medical equipment, prosthetics, orthotics and supplies public use file: a methodological overview. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/DME_Methodology.pdf
34. CMS: National Provider Identifier Standard (NPI). https://www.cms.gov/Regulations-and-Guidance/Administrative-Simplification/NationalProvIdentStand/

35. CMS: Medicare Provider Utilization and Payment Data. Physician and other supplier. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Physician-and-Other-Supplier.html
36. CMS: HCPCS-General Information. https://www.cms.gov/Medicare/Coding/MedHCPCSGenInfo/index.html
37. CMS: Medicare Provider Utilization and Payment Data: Part D Prescriber. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Part-D-Prescriber.html
38. CMS: Medicare Provider Utilization and Payment Data. Referring durable medical equipment, prosthetics, orthotics and supplies. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/DME.html
39. OIG: Office of Inspector General LEIE Downloadable Databases. https://oig.hhs.gov/exclusions/authorities.asp
40. OIG: Office of Inspector General Exclusion Authorities US Department of Health and Human Services. https://oig.hhs.gov/
41. Pande V, Maas W. Physician medicare fraud: characteristics and consequences. Int J Pharm Healthc Mark. 2013;7(1):8–33.
42. Bauder RA, Khoshgoftaar TM. 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). New York: IEEE; 2016. 11–19
43. Herland M, Khoshgoftaar TM, Bauder RA. Big data fraud detection using multiple medicare data sources. J Big Data. 2018;5(1):29.
44. Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A. Mining data with rare events: a case study. In: 19th IEEE international conference on tools with artificial intelligence (ICTAI 2007). New York: IEEE; 2007; vol. 2, p. 132–139
45. Herland M, Khoshgoftaar TM, Wald R. A review of data mining using big data in health informatics. J Big data. 2014;1(1):2.
46. Hu Q, Yu D, Xie Z. Neighborhood classifiers. Expert Syst Appl. 2008;34(2):866–76.
47. Bauder RA, Khoshgoftaar TM. The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data. Health inf Sci Syst. 2018;6(1):9.
48. Quinlan JR. C4. 5: Programs for machine learning. Amsterdam: Elsevier; 2014.
49. Breiman L. Manual on setting up, using, and understanding random forests v3. 1. Berkeley: Statistics Department University of California Berkeley; 2002.
50. Le Cessie S, Van Houwelingen JC. Ridge estimators in logistic regression. J Royal Stat Soc. 1992;41(1):191–201.
51. Chang C-C, Lin C-J. Libsvm: a library for support vector machines. ACM Trans Intel Syst Technol. 2011;2(3):27.
52. Khoshgoftaar TM, Seiffert C, Van Hulse J, Napolitano A, Folleco A. Learning with limited minority class data. In: Sixth international conference on machine learning and applications (ICMLA 2007). New York: IEEE; 2007. 348–353
53. Hasanin T, Khoshgoftaar TM. The effects of random undersampling with simulated class imbalance for big data. In: 2018 IEEE international conference on information reuse and integration (IRI). New York: IEEE; 2018. 70–79
54. Hasanin T, Khoshgoftaar TM, Leevy J, Seliya N. Investigating random undersampling and feature selection on bioinformatics big data. In: 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService). New York: IEEE; 2019. 346–356
55. Van Hulse J, Khoshgoftaar TM, Napolitano A. An empirical comparison of repetitive undersampling techniques. In: 2009 IEEE International Conference on Information Reuse & Integration. New York: IEEE; 2009. 29–34
56. Tukey JW. Comparing individual means in the analysis of variance. Biometrics. 1949;5:99–114.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

▶ Convenient online submission

▶ Rigorous peer review

▶ Open access: articles freely available online

▶ High visibility within the field

▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com