

A benchmark of health insurance fraud detection using machine learning techniques

Ossama Cherkaoui, Houda Anoun, Abderrahim Maizate

RITM Laboratory, CED Engineering Sciences, Hassan II University of Casablanca, Casablanca, Morocco

Article Info

Article history:

Received Mar 27, 2023

Revised Oct 27, 2023

Accepted Dec 2, 2023

Keywords:

Anomaly detection

Fraud detection

Health insurance fraud

Machine learning

Supervised classification

ABSTRACT

Health insurance fraud is a complex problem that also has a significant financial impact. Recently, with the availability of large volumes of data and the evolution of computing power, machine learning techniques have become the preferred method for fraud detection. However, the main difficulty facing researchers in this field is the lack of real data sets and the absence of reliable fraud labels. Most published studies use aggregated provider-level or simulated data to test fraud detection algorithms, which may not deliver accurate results. The present study aims to provide a more accurate assessment of fraud detection methods by using real detailed health insurance claims data to compare six of the most common supervised classification algorithms including neural networks and the use of two categorical feature preparation methods. The study was conducted under the guidance of insurance experts, who provided the fraud label inference rules and reviewed the results. A comprehensive description of the benchmarking process and an interpretation of the results are provided in this paper. The results show that supervised classification can be used effectively to detect health insurance fraud, improving detection accuracy by a factor of 4.2 (84% recall for a positive rate of 20%).

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ossama Cherkaoui

RITM Laboratory, CED Engineering Sciences, Hassan II University of Casablanca

Casablanca, Morocco

Email: ossama.cherkaoui@outlook.com

1. INTRODUCTION

Insurance fraud is a major issue that has important financial and business impacts. Gee and Button [1] covering 33 health organizations from 7 countries, the average loss of expenditure caused by fraud between 1997 and 2013 is estimated to be 6.19%, which represents, according to the same study, a loss in global healthcare expenditure of \$455 billion (€350 billion) in 2013 alone. Every insurance company has implemented methods and processes to detect fraud that are more or less sophisticated, but most of them lack clear visibility on the extent of the issue. Due to the generalization of health coverage in developed countries, as well as the rising number of claims and pressing deadlines, manual processing and investigation by auditors are no longer efficient. Thus, machine learning techniques have become widely used by insurance companies, analyzing large amounts of claims data to identify potential fraud cases. This paper presents a comparison of six machine learning algorithms, including neural networks and two categorical feature preparation methods, which are implemented and tested on real data from an insurance company operating in Africa.

Many papers in the literature address the application of machine learning techniques to health insurance fraud detection. However, most of the papers we examined focus on provider fraud using aggregated claims data. Indeed, it is very difficult to access detailed claims data at the patient level due to the constraints of data protection regulations. Bauder and Khoshgoftaar [2] performed a comparative study with

supervised, unsupervised, and hybrid machine learning approaches using four performance metrics and class imbalance reduction via oversampling and 80-20 under-sampling method. Results show that the 80-20 under sampling method demonstrates the best performance across learners. Furthermore, supervised methods such as random forests (RF), naive Bayes (NB), gradient boost machine, and deep neural networks (DNN) performed better than unsupervised or hybrid methods. In a different study, Bauder and Khoshgoftaar [3] conducted an empirical analysis of several unsupervised machine learning methods to detect outliers, indicating fraudulent medical providers, using the medicare part B big dataset [4]. They used receiver operating characteristic (ROC) curve and area under the curve (AUC) metrics to evaluate algorithms' performance. Results show a relatively good discriminative performance of the local outlier factor (LOF) [5] with AUC equal to 0.63. However, the AUC measure is not suitable for evaluating algorithm performance in the case of highly unbalanced datasets. On the other hand, the AUC precision-recall curve provides a better assessment, as it is sensitive to the minority class (cases of fraud).

Other types of algorithms have been proposed in the literature to detect health insurance fraud, each focusing on a specific type of fraud. Xu *et al.* [6] apply the PageRank algorithm to the Medicare-B dataset [7] to identify medical providers who prescribed significantly different medical procedures than other providers with the same specialties. Full Bayesian inference using probabilistic programming was used in [8] to detect outliers in Medicare datasets by generating payment probabilities based on provider specialties. The researchers used ensemble methods combining clustering, association rule mining, and support vector machines (SVM) to detect the most frequent and outlier patterns in claims data [9], [10].

The main difficulty facing researchers in the field of health insurance fraud is the lack of real data sets and the absence of reliable fraud labels. Indeed, most studies in the literature use aggregated data at the provider level [2], [3], [11], [12] or simulated data [13]. The current study provides a detailed comparison of the six most common supervised machine learning algorithms applied to real detailed health insurance claims data. Fraud labels were inferred using rules proposed by domain experts and based on reimbursement decisions, which are available and fully reliable. These labels capture auditor knowledge that can be applied to new, larger-scale data using machine learning algorithms to help identify health insurance abuse. The results were also reviewed by insurance auditors. The contributions of this paper are: i) providing a detailed comparison of six machine learning algorithms for fraud detection applied on a real insurance dataset where the results were validated by domain experts; ii) showing that supervised classification algorithms can achieve good accuracy on the basis of fraud labels deduced from auditors' historical reimbursement decisions; and iii) offering a comparison of categorical embedding and one-hot encoding data preparation methods.

This paper is structured as follows: section 2 presents a high-level description of the fraud detection process and highlights some specific features of health insurance datasets, followed by an overview of the main machine learning algorithms used for fraud detection. In section 3, a more detailed description of the data used for testing, data preparation methods, algorithm hyperparameter tuning process, and evaluation metrics is provided. The findings of the benchmark tests and their analysis in terms of business interpretation are presented in section 4. Section 5 concludes by identifying unresolved issues not covered in this study and prospects for future research.

2. HEALTH INSURANCE FRAUD DETECTION PROCESS

The fraud detection process in the health insurance domain relies on the analysis of historical claims to identify unusual patterns or anomalies and generate scores for each instance to indicate the likelihood of fraud. Knowing that final confirmation of fraud can only be done with a further investigation by domain experts, the main goal of the fraud detection process is to limit the scope of work for investigators so that they can focus on the most likely fraud cases. The schema in Figure 1 is a high-level representation of the fraud detection process.

Fraud in the health insurance domain comes in different types depending on the fraudster's role and the data that was altered. To our knowledge, there is no general fraud detection method that can detect all fraud types, so targeting a specific type in advance is required to prepare the data accordingly. For example, if the goal is to detect fraud committed by service providers, then the data should be aggregated at provider levels and specify fraud labels by the provider.

Reliable fraud labels are generally not available in insurance claims datasets. Evidently, labels are required for the training phase of classification algorithms, but they are also required to evaluate and compare the performance of unsupervised anomaly detection algorithms. The predominance of categorical attributes that are generally very correlated (diagnostic, provider specialty, patient gender, and treatment) is another characteristic of health insurance datasets. On the other side, most of the common machine learning algorithms require numerical inputs, so it is necessary to transform categorical attributes accordingly.

Many methods are available to achieve this goal, the most common being one-hot encoding and ordinal encoding when the order of class labels is relevant. For neural networks, it has been proven that an embedding layer to process input categorical features gives better results than one-hot encoding [14]. Both methods, one-hot encoding, and categorical embedding, are tested and compared in the present study. Finally, the tests conducted in this benchmark show that the depth of the historical data and the splitting method-random or based on a cut-off date-used during the training phase of the classification methods can greatly influence the accuracy of the results. Indeed, business rules, prices and fraudsters' practices evolve over time, so a long data history does not necessarily improve fraud detection, as it is irrelevant to mix different data related to completely different contexts.

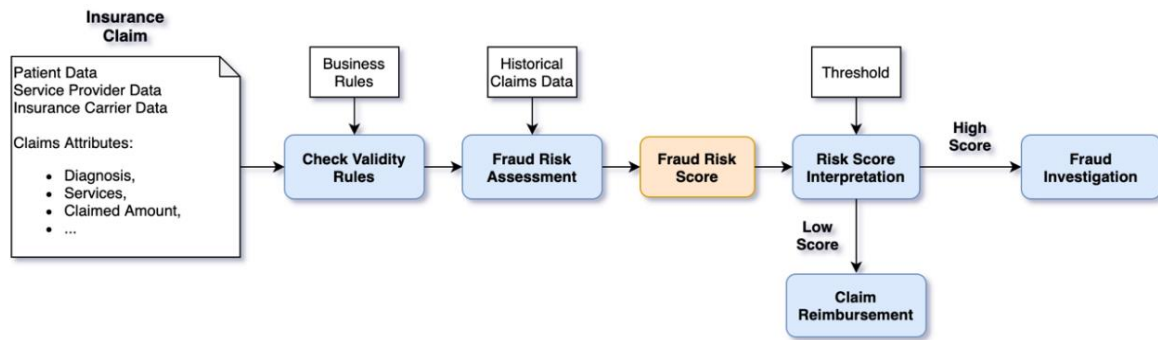


Figure 1. High-level insurance fraud detection process

2.1. Fraud detection algorithms

Fraud detection is addressed by two families of machine learning algorithms: unsupervised anomaly detection and classification. Unsupervised models look for the underlying structure and patterns in the data to identify anomalies. They rely on assumptions to define what an anomaly is by using different criteria such as distances, densities, and probabilities. On the other hand, classification models rely only on the fraud labels to support the training process and make predictions. Evaluating the performance of both types of algorithms: unsupervised and classification, requires fraud labels, which in most cases are not available or are not reliable. The main algorithms in each family are presented in the rest of this section.

2.1.1. Unsupervised anomaly detection

The goal of unsupervised anomaly detection is to calculate a score for each data instance to indicate the likelihood of fraud. Anomaly scores reflect specific criteria used by the algorithm to define anomalous instances. To extract predicted labels (anomaly or normal), a threshold is used and compared to generated scores. However, it is not easy to interpret scores and choose the right threshold to use, as it is not always consistent across datasets and depends on the normalization method applied to data.

Many algorithms for unsupervised anomaly detection have been proposed in the literature. Many reviews [6], [8], [15] provide a description of unsupervised anomaly detection algorithms and grouping them into the following main categories: i) nearest neighbors based, ii) tree based, iii) clustering based, iv) statistical, v) probability based, vi) subspace based, and vii) deep learning based. Nearest neighbors based methods: anomaly detection based on nearest neighbors relies on the assumption that anomalies are isolated in the dataset according to a certain distance metric, and will therefore have fewer instances in their neighborhood than normal instances. Examples of algorithms based on nearest neighbors include the k-nearest neighbors (KNN) anomaly detection method, which defines the score of an anomaly as the average of the distances to the KNN, the LOF method [5], which uses relative density to calculate the anomaly score, and numerous other variants of LOF that have been proposed to improve its performance and efficiency for specific types of dataset [8].

Tree based methods: the isolation forest algorithm proposed in [16] introduces a new family of tree-based anomaly detection. It assumes that anomalies can be easily isolated using a randomly constructed tree, called an isolation tree, fitted to the data. This means that a reduced number of edges in the isolation tree will be traversed to reach the anomalous instances. Consequently, the isolation forest algorithm defines an anomaly score that is inversely correlated with the average path length to reach an instance in the isolation tree. Numerous variants have been proposed to extend isolation forest, including extended isolation forest [17] and k-means-based isolation forest [18].

Clustering based methods: clustering based anomaly detection defines anomalies as instances that do not belong to any of the identified clusters, or are far from them [8]. This is only possible if the clustering algorithm generates flexible membership scores or allows certain instances (i.e. anomalies) to belong to no cluster at all. Known clustering algorithms can be used for this purpose, but they may not be effective in identifying anomalies, as they are not designed for anomaly detection in the first place. The anomaly detection process presented in [7] uses a variant of k-means clustering that generates dynamic clusters and then identifies anomalies using minimum spanning trees (MST). Other variants of k-means have been proposed to improve its robustness and perform both clustering and anomaly detection tasks, for example: k-means [9] and k-means clustering with outlier removal (KMOR) [10]. The cluster-based local outlier factor (CBLOF) [19] is a local anomaly detection method that can be based on any clustering algorithm. CBLOF distinguishes between large and small clusters to define anomaly scores combining cluster size and distances from clusters centers.

Statistical methods: statistical anomaly detection assumes that the data follow an underlying probability distribution, and that anomalies lie in areas of low probability. Parametric statistical methods infer the parameters of the data distribution, then generate anomaly scores using the probability density function or a statistical test. Gaussian model is an example of statistical anomaly detection, which assume that the data is generated from a Gaussian distribution and uses maximum likelihood estimation (MLE) to infer its mean and variance. Any instance that deviates from the mean beyond a predefined threshold is considered an anomaly. Non-parametric statistical methods do not assume a specific distribution for the data. These methods use histograms or kernel density estimation to estimate the probability density function and generate anomaly scores. An example of a non-parametric statistical algorithm for anomaly detection is histogram based outlier score (HBOS) [20], which uses histograms for each feature of the input data to estimate the density under the assumption that they are statistically independent. HBOS has the advantage of supporting mixed (numerical and categorical) data, and is highly efficient for large datasets.

Subspace based methods: subspace-based anomaly detection assumes that there is a subspace of data in which anomalies are more clearly identified than normal cases. Anomaly detection process therefore involves first reducing the dimensionality of the data, then identifying anomalies in the resulting subspace. Shyu *et al.* [21] proposes an anomaly detection method using robust principal component analysis to reduce the dimensionality of the data, and taking into account only the scores of the major and minor components. For high-dimensional data, priority is given to the efficiency of anomaly detection and to overcoming the problems associated with the curse of dimensionality [22]. The projection indexed nearest-neighbours (PINN) algorithm [23] uses distance-preserving random projection to find the nearest neighbors in a pre-processing step, then calculates LOF score using the exact distances in the original space. Locality-sensitive outlier detection (LSOD) [24] uses locality-sensitive hashing as a pre-processing step to rank the dataset instances before performing an approximate nearest-neighbor search to calculate the anomaly score.

Probability based methods: probability based anomaly detection assumes a generative model of the data and uses probability inference techniques to learn its parameters. Like statistical methods, probability based anomaly detection defines anomalies as instances having a low probability according to the assumed model. Examples of this approach are presented in [11], [13].

Deep learning based methods: deep learning methods have many applications in the field of anomaly detection. They can be used for feature extraction, feature representation learning and end-to-end anomaly score learning [25]. Replicator neural networks (RNN) [26] is the earliest anomaly detection deep learning method. The RNN concept involves learning a representation of the data using a three hidden-layer perceptron neural network that is trained to reproduce the input by minimizing the reconstruction error. The RNN assumes that anomalies will have a larger reconstruction error than normal cases. The same approach is proposed by autoencoders [27], which use a different neural network structure to learn a representation of the data in a latent space of lower dimension. Like RNNs, autoencoders use the reconstruction error as an anomaly score. Other types of neural network have been proposed for anomaly detection; a detailed review is available in [25].

2.1.2. Classification

Fraud detection can be considered a special case of binary classification with a very unbalanced distribution. Therefore, using classification algorithms for fraud detection requires additional data preparation and hyperparameter tuning to balance the dataset using oversampling, undersampling or weighted cost methods to put more costs on anomaly cases. However, the main limitation of classification for anomaly detection is the lack of reliable labels, which leads to a high rate of false negatives in the dataset. In addition, classification learns to identify anomalies on the basis of historical cases in the training set, so its ability to detect totally new fraud patterns will be very limited. Classification algorithms are the subject of extensive research in the literature. In the current benchmark, tests are limited to five algorithms belonging to five

different classification families: logistic regression (LR) [28], extreme gradient boosting (XGBoost) [29], RF [30], SVM [31], and neural networks [32].

3. CASE STUDY PRESENTATION

This paper presents a real case study on fraud detection in a health insurance company operating in many countries, which is referred to as “the insurance” in the rest of the paper. Health insurance abuse is a major concern for the insurance. Moreover, it has no visibility on the extent of the problem and related losses. The audit department is responsible for controlling and checking claims validity rules and investigating any potential fraud cases. Auditors work when necessary to identify potential fraud cases, which is a very challenging task considering that the number of submitted claims can be more than 20,000 per month in one country alone. The goal of this study was to provide auditors with a solution that will identify most likely fraud and abuse cases to limit the scope of work and increase efficiency. Six classification methods were compared in this benchmark: LR [28], XGBoost [29], RF [30], SVM classification [31], neural network with one-hot encoding (NN-OHE), and neural network with entity embedding (NN-EE) [14], [32]. Implementation is based on scikit-learn [33], XGBoost, and PyTorch [34] libraries.

3.1. Data preparation

The data corresponds to the history of patient claims and reimbursement decisions for two years (2018 and 2019) in a country where the insurance operates with an average number of claims per month equal to 17,000. Each data instance corresponds to an individual claim and is composed of categorical, numerical, and date features. Table 1 lists all features that were used for fraud detection.

3.1.1. Features transformation

Different transformation methods have been applied to the data, depending on the type of features. For categorical features, the one-shot encoding transformation is used, and entity embedding for the neural network algorithm. For numeric features, the standard scaling transformation is used. Date features are extracted with the month as a categorical feature and the day of the year as a numerical feature, after being divided by 366 to normalize the values. A summary of transformation methods for each feature is illustrated in Table 2.

Table 1. Claims features used for fraud detection

Attribute	Type	No of classes/range
Pathology code	Categorical	1369
Medical intervention code	Categorical	635
Health provider specialization	Categorical	17
Patient relationship with subscriber	Categorical	5
Reimbursement type	Categorical	3
Medical intervention date	Date	5-Jan-2018/31-Dec-2019
Claimed amount	Numerical	0.01/27,800,192
Reimbursed amount	Numerical	0.01/27,800,192

Table 2. Features' transformation methods

Attribute	Transformation method
Pathology code	One-hot encoding, entity embedding
Medical intervention code	One-hot encoding, entity embedding
Health provider specialization	One-hot encoding, entity embedding
Patient relationship with subscriber	One-hot encoding, entity embedding
Reimbursement type	One-hot encoding, entity embedding
Medical intervention date	One-hot encoding, entity embedding
Claimed amount	Normalization by dividing by 366
Reimbursed amount	Standard scaling

3.1.2. Entity embedding for neural networks

Entity embedding is a special neural network layer that uses different weights for each class of input categorical feature. The goal is to transform categorical features into relevant continuous numerical features. Entity embedding was successfully tested in many cases and is proven to be better than one-hot encoding, which is not adapted for neural networks [14].

Figure 2 shows the neural network model used for the benchmark tests. It includes an embedding layer for each categorical feature to map classes to weight vectors. The outputs of embedding layers are concatenated with transformed numerical features to be processed by three fully connected neural network

layers using the rectified linear unit (ReLU) activation function. The final layer of the neural network returns a numerical value which is used to generate scores (probabilities) and predictions using the sigmoid function, and to calculate classification metrics such as mean accuracy and the precision-recall curve. The same output is used to calculate loss with the binary cross-entropy method using different weights for normal and positive classes (3 times weight ratio) to compensate for dataset imbalance. In the present study, the same neural network illustrated in Figure 2 is tested but without entity embedding in order to assess the effect of the categorical feature transformation method on classification performance. For this second neural network, categorical features are transformed using the one-hot encoding method.

Figures 3(a) and (b) shows the evolution of training and validation loss per epoch for the neural network with feature embedding and one-hot encoding. In both cases, the loss decreases during the training and validation phases, and the minimum loss in the validation data is reached after only 25 epochs. Training loss continues to decrease after 25 epochs, but this is due to model overfitting and brings no benefit to generalization. Further tests were carried out with different neural network configurations, but the use of more than 8 nodes per layer resulted in an over-fitting of the training dataset with no benefit to test performance.

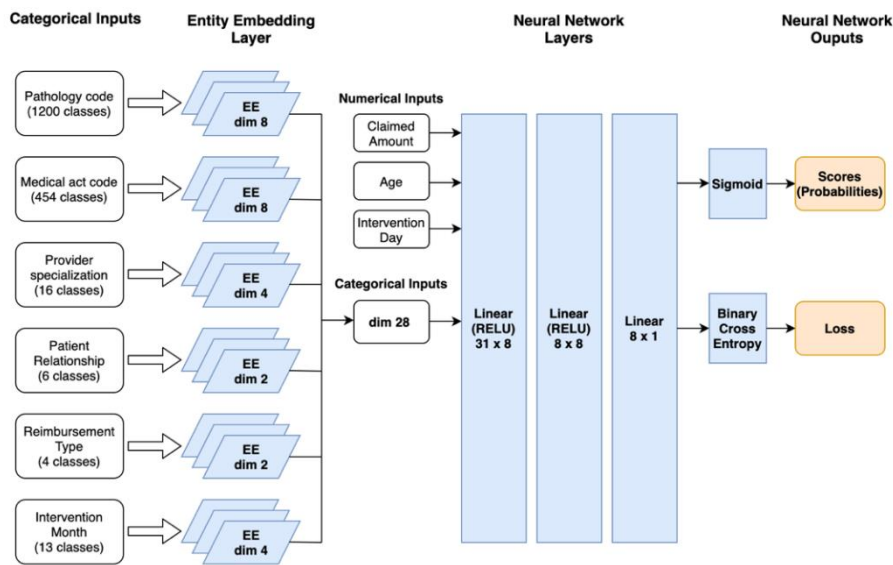


Figure 2. Classification neural network with entity embedding layers

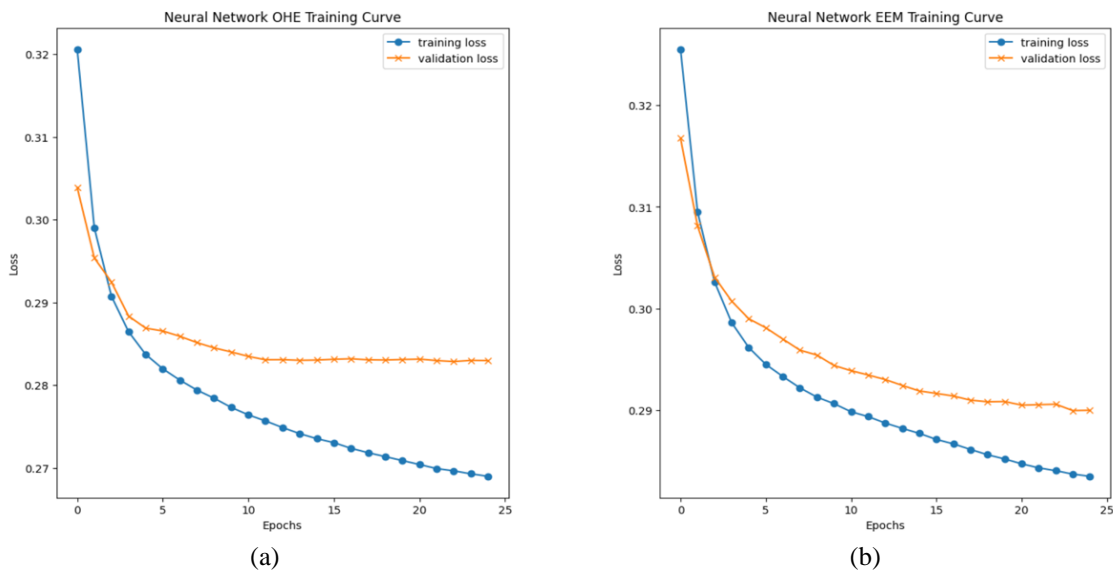


Figure 3. Neural networks training loss curves for (a) one hot encoding and (b) entity embedding

3.2. Algorithms hyperparameter tuning

The hyperparameter tuning step aims to set the optimal parameters for each tested algorithm. It was performed with a 10-fold cross-validation process using only the training dataset. Different hyperparameter values were tested for each algorithm, then the best values were chosen using the Matthews correlation coefficient (MCC), which represents a compromise between precision and recall, and produces a more informative and accurate score in the evaluation of binary classifications than precision and F1 score [35]. To address the imbalance between normal and anomaly classes, a higher cost has been assigned to anomaly instances in the algorithm parameters. Table 3 presents the optimal hyperparameters after the tuning.

Table 3. Optimal hyperparameters and MCC scores after cross-validation

Model	MCC score	Best parameter values	
		Parameter	Value
LR	0.167	Inverse of regularization strength	0.1
		Classes weights	Balanced
		Maximum number of iterations	10000
XGBoost	0.361	Boosting learning rate	0.3
		Maximum tree depth for base learners	5
		Number of boosting rounds	200
		Balancing of positive and negative weights	3
RF	0.282	Weights associated with classes	Balanced
		The function to measure the quality of a split	Entropy
		The maximum depth of the tree	None
		The minimum number of samples required to be at a leaf node	1
SVM	0.231	Inverse of regularization strength	1
		Classes weights	Balanced
		kernel type to be used in the algorithm	RBF

3.3. Evaluation metrics

In order to compare algorithm performance on the test dataset, the MCC metric calculated during the hyperparameter setting stage cannot be used, as it depends on the default decision function of each algorithm, which may use different thresholds corresponding to different positive rates. The commonly used evaluation metric for classification is the ROC curve showing the performance of a classification model at all thresholds and the area under the ROC curve [36]. However, the AUC can produce inaccurate results if the dataset is unbalanced, particularly when the main objective is to predict frauds which constitute the minority class. In this benchmark, the evaluation metric used is the average precision score based on the precision-recall curve [36], which is better suited to anomaly detection algorithms.

4. RESULTS AND DISCUSSION

Figure 4 shows the precision-recall curves and average precision for the algorithms tested. The results show good performance of XGBoost and RF algorithms with similar average precision, with values at 0.37 and 0.32 respectively. Neural network algorithms with one-hot encoding and with entity embedding have similar average precisions at 0.21 and 0.18 respectively. SVM with radial basis function kernel (RBF) algorithm has 0.18 average precision and LR has the lowest performance with 0.13 average precision. One can notice in precision-recall curves that some algorithms have the same behavior: RF and XGBoost both based on decision trees have very similar curves, and neural networks with either one-hot encoding or with entity embedding have almost identical curves. On the other hand, the results show that entity embedding transformation doesn't provide any benefit for the neural network algorithm in our case. Indeed, using the one-hot encoding transformation, the same average precision was achieved.

4.1. Results interpretation from a business perspective

From a business perspective, the fraud detection solution is a tool to prioritize investigators' efforts to focus on the most suspicious cases. The insurance has a time service level agreement to process claims but also has a limited capacity to review more than 20,000 claims per month. Auditors need an indicator that will aid them in deciding which claims to review while ensuring a high accuracy. Therefore, the auditors would be more interested in the performance of fraud detection within the top N potential fraud cases that they can investigate. This can be measured by the $Recall@N$ evaluation metric which is one of the metrics used to evaluate recommender systems [37]. $Recall@N$ calculates the recall considering only the N instances with highest scores, it represents the fraction of real fraud cases in the top results returned by anomaly detection algorithm. In the current study, recall at a specific positive rate: $Recall@PR$ is used instead of $Recall@N$. For

example, if the number of instances in the test dataset is 40,000, then a positive rate of 0.2 means that the algorithm will make a positive prediction for the top 20% instances ($N=8,000$) with the highest scores.

If fraud cases are randomly predicted, the *Recall@PR* should be equal to the positive rate. Consequently, the gain obtained by using the fraud detection algorithm can be defined as the ratio between the *Recall@PR* and the positive rate.

$$Gain = \frac{Recall@PR}{Positive\ Rate}$$

The insurance can choose the minimum number of claims to be reviewed monthly to achieve a specific target gain. As the insurance auditors examine more claims, they will detect more cases of fraud, and the gain will improve. However, the false-positive rate will also be higher, resulting in lower precision. Table 4 shows the performance metrics and gains for specific false positive rates for each algorithm. XGBoost shows the best results overall. At 20% positive rate, XGBoost achieves a gain of 4.20, this means that the insurance improves fraud detection by a factor of 4.2 and detects 84% of fraud cases by investigating only 20% of claims with the highest scores.

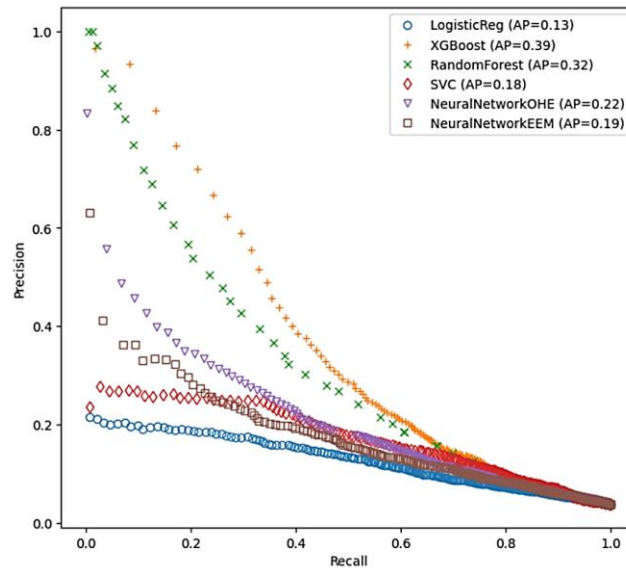


Figure 4. Precision-recall curves for tested algorithms

Table 4. Algorithms’ performance metrics at specific detection rates

		LR	XGBoost	RF	SVM	NN-OHE	NN-EE
5% positive rate	Gain	4.80	8.80	8.00	6.60	6.60	6.00
	Recall	0.24	0.44	0.40	0.33	0.33	0.30
	Precision	0.19	0.34	0.31	0.25	0.25	0.23
	MCC	0.18	0.36	0.33	0.25	0.25	0.23
	F1	0.21	0.38	0.35	0.28	0.28	0.26
10% positive rate	Gain	3.90	5.70	5.40	4.90	4.70	4.60
	Recall	0.39	0.57	0.54	0.49	0.47	0.46
	Precision	0.15	0.22	0.22	0.19	0.18	0.17
	MCC	0.19	0.31	0.31	0.26	0.25	0.24
	F1	0.22	0.32	0.31	0.27	0.26	0.25
15% positive rate	Gain	3.93	4.67	4.60	4.53	4.40	4.40
	Recall	0.59	0.70	0.69	0.68	0.66	0.66
	Precision	0.11	0.13	0.14	0.13	0.12	0.12
	MCC	0.20	0.25	0.25	0.24	0.23	0.23
	F1	0.19	0.22	0.23	0.22	0.21	0.21
20% positive rate	Gain	4.00	4.20	4.00	4.20	4.10	4.10
	Recall	0.80	0.84	0.80	0.84	0.82	0.82
	Precision	0.08	0.08	0.09	0.08	0.08	0.08
	MCC	0.16	0.18	0.18	0.18	0.17	0.17
	F1	0.14	0.15	0.15	0.15	0.14	0.14

5. CONCLUSION

The current study shows encouraging results for supervised machine learning methods in fraud detection. Indeed, the XGBoost algorithm achieved a recall of 84% for a positivity rate of 20%. This represents an improvement of the capacity of the insurance to detect fraud by a factor of 4.2 (gain=4.2). However, the processing of categorical variables in the current study was limited to two methods: one-hot encoding and entity embedding, which did not provide all the valuable information that could improve the accuracy and interpretability of fraud detection. One-hot encoding and entity embedding methods are not sufficient to prepare categorical variables for fraud detection and need to be complemented by other techniques to enrich the input data for machine learning algorithms. In future work, the focus will be on ensemble methods combining dimension reduction techniques and a probabilistic approach that are suitable for both numerical and categorical features to address aspects that have not been covered in the current study, including: i) detection of relationships and hidden patterns in categorical application variables, such as associations and correlations; ii) analysis of temporal (frequency of services) and spatial information on requests (location of patients and service providers); iii) correlation of claims submitted by patients belonging to the same groups (family or company); and iv) the business interpretation of fraud detection results based on a combined visual representation of categorical and numerical attributes.




REFERENCES

- [1] J. Gee and M. Button, *The financial cost of healthcare fraud 2015: What data from around the world shows*, London: Forensic & Counter Fraud Service, 2015.
- [2] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using machine learning methods," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun Mexico, 2017, vol. 2017, pp. 858–865, doi: 10.1109/ICMLA.2017.00-48.
- [3] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using random forest with class imbalanced big data," in *2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018*, Salt Lake City, UT, USA, 2018, pp. 80–87, doi: 10.1109/IRI.2018.00019.
- [4] "Medicare program-general information," *U.S. Centers for Medicare & Medicaid Services*. 2023. Accessed: Apr. 09, 2023. [Online]. Available: <https://www.cms.gov/about-cms/what-we-do/medicare>.
- [5] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *SIGMOD 2000- Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 93–104, doi: 10.1145/342009.335388.
- [6] X. Xu, H. Liu, and M. Yao, "Recent progress of anomaly detection," *Complexity*, vol. 2019, 2019, pp. 1–12, doi: 10.1155/2019/2686378.
- [7] M. F. Jiang, S. S. Tseng, and C. M. Su, "Two-phase clustering process for outliers detection," *Pattern Recognition Letters*, vol. 22, no. 6-7, pp. 691–700, 2001, doi: 10.1016/S0167-8655(00)00131-8.
- [8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009, doi: 10.1145/1541880.1541882.
- [9] S. Chawla and A. Gionis, "k-means-: A unified approach to clustering and outlier detection," in *Proceedings of the 2013 SIAM International Conference on Data Mining, SDM 2013*, 2013, pp. 189–197, doi: 10.1137/1.9781611972832.21.
- [10] G. Gan and M. K. P. Ng, "K-Means clustering with outlier removal," *Pattern Recognition Letters*, vol. 90, pp. 8–14, 2017, doi: 10.1016/j.patrec.2017.03.008.
- [11] 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 2016*, Anaheim, CA, USA 2016, pp. 347–354, doi: 10.1109/ICMLA.2016.0063.
- [12] J. Seo and O. Mendelevitch, "Identifying frauds and anomalies in Medicare-B dataset," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jeju, Korea (South), 2017, pp. 3664–3667, doi: 10.1109/EMBC.2017.8037652.
- [13] T. Ekina, F. Leva, F. Ruggeri, and R. Soyer, "Applications of bayesian methods in detection of healthcare frauds," *Chemical Engineering Transaction*, vol. 33, pp. 151–156, 2013, doi: 10.3303/CET1333026.
- [14] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," *arXiv-Computer Science*, pp. 1–9, 2016.
- [15] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, pp. 1–31, 2016, doi: 10.1371/journal.pone.0152173.
- [16] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012, doi: 10.1145/2133360.2133363.
- [17] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1479–1489, 2021, doi: 10.1109/TKDE.2019.2947676.
- [18] P. Karczmarek, A. Kiersztyn, W. Pedrycz, and E. Al, "K-Means-based isolation forest," *Knowledge-Based Systems*, vol. 195, pp. 1–15, 2020, doi: 10.1016/j.knosys.2020.105659.
- [19] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1641–1650, 2003, doi: 10.1016/S0167-8655(03)00003-5.
- [20] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, no. 1, pp. 59–63, 2012.
- [21] M.-L. Shyu, S.-C. Chen, K. Sarinapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *IEEE foundations and new directions of data mining workshop in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03)*, 2003, pp. 172–179.
- [22] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*, vol. 5, no. 5, pp. 363–387, 2012, doi: 10.1002/sam.11161.
- [23] T. De Vries, S. Chawla, and M. E. Houle, "Finding local anomalies in very high dimensional space," in *2010 IEEE International Conference on Data Mining, ICDM, Sydney, NSW, Australia, 2010*, pp. 128–137, doi: 10.1109/ICDM.2010.151.




- [24] Y. Wang, S. Parthasarathy, and S. Tatikonda, "Locality sensitive outlier detection: A ranking driven approach," in *2011 IEEE 27th International Conference on Data Engineering*, Hannover, Germany, 2011, pp. 410–421, doi: 10.1109/ICDE.2011.5767852.
- [25] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021, doi: 10.1145/3439950.
- [26] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Data Warehousing and Knowledge Discovery*, Berlin, Heidelberg: Springer, pp. 170–180, 2002, doi: 10.1007/3-540-46145-0_17.
- [27] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," in *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, Cham: Springer, pp. 353–374, 2023.
- [28] D. G. Kleinbaum and M. Klein, "Introduction to logistic regression," in *Logistic Regression*, New York: Springer, 2010, pp. 1–39, doi: 10.1007/978-1-4419-1742-3_1.
- [29] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [30] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995, doi: 10.1007/BF00994018.
- [32] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, pp. 1–41, 2018, doi: 10.1016/j.heliyon.2018.e00938.
- [33] F. Pedregosa *et al.*, "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [34] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019, pp. 8026–8037.
- [35] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.
- [36] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining and Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015, doi: 10.5121/ijdkp.2015.5201.
- [37] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2019, doi: 10.1007/s13042-017-0762-9.

BIOGRAPHIES OF AUTHORS






Ossama Cherkaoui    is a doctoral student at Hassan II University in Morocco. In 1997, he obtained from Ecole Nationale d'Informatique et d'Analyse des Systèmes (ENSIAS) a computer science engineer degree. His research areas are anomaly detection algorithms and the application of machine learning techniques for fraud detection in healthcare insurance. He can be contacted at email: ossama.cherkaoui@outlook.com.



Houda Anoun    obtained her Ph.D. in computational linguistics from the University of Bordeaux I in 2007, and her engineering degree in software engineering from ENSEIRB Bordeaux in 2003. Currently, she is a professor in the Department of Computer Science at the Ecole Supérieure de Technologie of Hassan II University since 2009. She is working in the field of AI, especially in big data, machine learning, and deep learning. She can be contacted at email: houda.anoun@gmail.com.



Abderrahim Maizate    received his Engineering Diploma in Computer Science from the Hassania School of Public Works since 2004 and DESA degree from ENSIAS in 2007. Then, he received his Ph.D. degree from the Chouaib Doukkali university. He is currently professor researcher at the Hassan II University, Casablanca, Morocco. His research interest includes wireless communication, mobile communication, wireless sensor networks, AI, and big-data. He can be contacted at email: maizate@hotmail.com.