Multiple Inputs Neural Networks for Fraud Detection

Mansour Zoubeirou a Mayaki, Michel Riveill

Multiple Inputs Neural Networks for Fraud Detection

Mansour Zoubeirou A Mayaki

Université Côte d'Azur
Inria, CNRS, Nice France
mansour.zoubeirou-a-mayaki@inria.fr

Michel Riveill

Université Côte d'Azur
CNRS, Inria, Nice France
michel.riveill@unice.fr

***Abstract*—This study aims to use artificial neural network based classifiers to predict fraud, particularly that related to health insurance. Medicare fraud results in considerable losses for governments and insurance companies and results in higher premiums from clients. Medicare fraud costs around 13 billion euros in Europe and between 21 billion and 71 billion US dollars per year in the United States. To detect medicare frauds, we propose a multiple inputs deep neural network based classifier with an autoencoder component. This architecture makes it possible to take into account many sources of data without mixing them and makes the classification task easier for the final model. We use the data sets from the Centers for Medicaid and Medicare Services (CMS) of the US federal government and four benchmarks fraud detection data sets. Our results show that although baseline artificial neural network give good performances, they are outperformed by our multiple inputs neural networks. We have shown that using an autoencoder to embed the provider behavior gives better results and makes the classifiers more robust to class imbalance.**

***Index Terms*—Medicare fraud detection, Anomaly detection, Imbalanced data, Machine learning, Deep neural networks**

I. Introduction

The progress made in the field of big data and data management makes it possible to fight fraud more effectively in several business sectors such as finance, banking and insurance. Detected and undetected fraud cost European customers and insurers around 13 billion euros per year. In the field of medicare, in France the compulsory scheme detected 261.2 million euros of fraudulent services in 2018, mainly due to medicare providers. In the United States, according to the Federal Bureau of Investigation (FBI), fraud represents $5 - 10\%$ of medicare claims and costs insurance companies between 21 billion and 71 billion per year. The most common types of fraud include billing for appointments that the patient has missed, billing for services that are more complex than those performed, or billing for services not provided.

Most insurance companies use business rule based fraud detection systems. These methods, although effective, are often very difficult to set up and maintain. Indeed, a rule-based fraud detection system constantly requires the presence of experts in the field and constant updates of the rules. Models based on statistical methods and machine learning make it possible to automatically build patterns and thus detect fraudulent activities effectively.

The main difficulty in applying machine learning techniques in fraud detection or more generally anomaly detection is that you don't have enough data labeled as anomalous or fraudulent. Thus, you end up in a situation of imbalanced class where one class is poorly represented compared to the others. For example in medicare, fraudulent transactions often represent less than 5% of all transactions. The high imbalance rate makes it very difficult for machine learning algorithms to learn as they will tend to favor the majority class. In this study we use publicly available medicare data sets from the Centers for Medicare and Medicaid Services (CMS) for period 2017–2019. Moreover, in order to evaluate our method's capabilities, we evaluate its performance on four publicly available bench mark data sets. The CMS data sets contain the hospitalization requests (Inpatient Data), the outpatient care requests (Outpatient Data) and the claims details. We also use the Office of Inspector General's list of excluded individuals and entities (LEIE). The LEIE table contains the list of healthcare providers excluded from the healthcare system for illegitimate or fraudulent activities. The main challenge working with this data set is that it is highly imbalanced with a fraud rate between 0.038% and 0.074%. Another challenge is that it exhibits big data properties. To detect medicare frauds, we propose a multiple inputs deep neural networks based classifier with an autoencoder component. We call this architecture **MINN-AE**. This architecture makes it possible to take into account many sources of data without mixing them and makes the classification task easier for the final model. The autoencoder part of MINN-AE plays a dimension reduction role for the provider data and its latent vector describes the provider behavior over time. The rest of the paper is outline as follows. The **Related works** section discusses the other studies and articles related to imbalance data handling, deep learning for anomaly and medicare fraud detection. In the third section, we describe our approach and the model's architecture. Section **Experimental design** is dedicated to the choice of hyperparameters and loss functions. The experimental data sets and pre-processing steps are described in the fifth. The results are presented and discussed in the last section.

II. Related works

The work presented here does not only concern research in the fields of medicare or fraud detection. We also present

some techniques proposed to remedy the problem of class imbalance. These two concepts are inseparable because in fraud detection we always face the problem of class imbalance.

A. Fraud Detection and Resampling Methods

The Centers for Medicare and Medicaid Services (CMS) data has been used in numerous studies to detect medicare fraud. Most of these studies use resampling techniques to overcome the imbalance class issue (Bauder et al. [1]; Liu et al. [2]; Herland et al. [3]; Johnson et al. [4]; Van et al. [5]). In there study, Herland et al. [3] show that the combination of the three parts of CMS data makes it possible to detect more precisely fraudulent activities. They compared the performances of logistic regression, random forest and gradient boosting classifiers on each part of the data taken separately with those obtained grouping all the parts and results show that the performance of all classifiers improves dramatically using all parts of the data, and that logistic regression outperforms all other models. Using the CMS data from 2010, Liu et al. [2] added some geo-location information to detect fraud. They went from the hypothesis that medicare beneficiaries are senior, disabled or poor and prefer to choose the health service providers locating in a relatively short distance and if the distance between the providers location and the client living place is too long, it may imply a fraud. Bauder et al. [1] used three different classifiers to detect fraudulent medicare provider claims: C4.5 decision tree (C4.5), Support Vector Machine (SVM), and Logistic Regression (LR). They used the CMS data over the period 2012-2015 combined with the Office of Inspector General's list of excluded individuals and entities (LEIE). The authors also used random undersampling technique to handle the class imbalance problem. In their study, Johnson et al. [4] compared six resampling techniques for imbalanced classes using the CMS data over the period 2012–2016. These authors combined artificial neural network models with class imbalance techniques to predict fraud. They tested random undersampling (RUS), random oversampling (ROS), mean square error (MSE) and Focal Loss techniques among others. According to their results, RUS improves the performance of the classification algorithm if the majority class share is above 99%. The authors then conclude that maintaining sufficient representation of the majority class is more important than reducing the level of class imbalance, and that down-sampling until classes are balanced can deteriorate classification performance. Van et al. [5], in another study also compared different resampling techniques using 11 types of classifiers. In their experiments, they used 35 different data sets with degrees of imbalance (ratio between the number of samples in the minority class and that of the majority class) varying between 1.33% and 35%. The resampling techniques used in this article are: random undersampling (RUS), random oversampling (ROS), one-sided selection (OSS), cluster-based oversampling (CBOS), Wilson's editing (WE), synthetic minority oversampling technic (SMOTE) and borderline-SMOTE (BSM).

Most of these studies come to the conclusion that undersampling (down-sampling) is more efficient than over-sampling. These results go against what one might have expected as undersampling often leads to a loss of information. One possible explanation is that in some situations, adding new artificial data will add more noise than useful information to the model. Depending on the complexity of the problem (or data), it is necessary to test the two approaches (down-sampling and over-sampling) to see which one fits best.

B. Algorithm Level Methods for Imbalanced Classes

To overcome the problem of class imbalance, some authors propose to alter the learning algorithm in the way that it takes into account the problem (Wang et al. [6]; Haishuai et al. [7]; Lin et al. [8]). The main idea of algorithm level method is to modify the learning algorithms so that they give more importance to the samples from the minority class which is often the class of interest.

Lin et al. [8] proposed an algorithm level method which consists in rewriting the classical entropy loss function by integrating two new parameters: α takes into account the imbalanced issue and γ (gamma) the complexity of classifying the samples. This new loss function called **Focal Loss** is obtained by multiplying the classical cross entropy (CE) by a modulation factor $\alpha(1-p)^{\gamma}$. hyperparameter $\gamma \geq 0$ adjusts the rate at which easy examples are down weighted and α is a class-wise weight used to give more importance to the minority class [9]. Lin et al. [8] applied their new cost function (Focal Loss) to object detection in images and their results show that this loss function gives better performance than most benchmark models. Wang et al. [6] proposed another algorithm level method called **mean false error** (MFE) which consist in decomposing the classical mean squared error (MSE) in two components in other to give more weights to the minority class samples. They rewrite the classical MSE as a kind of weighted average of the errors of the two classes. In this way, all the classes participate equally in the final loss function. Haishuai et al. [7] in their paper used an artificial neural network based model with a **cost matrix** to predict readmission of patients from a hospital. They defined a **cost matrix** such that the cost of misclassified readmission (False negative) is greater than that of misclassified non-readmission (false positive). This technique can be seen as an algorithm level method because during optimization, the model will tend to penalize more or give more weight to the minority class (readmission) samples in the loss function.

Algorithm level methods often give better results than resampling methods as they don't alter the training data and don't lead to a loss of information. However in some situations, when you don't have enough data, oversampling can be a good way to extend your data set. Moreover, when the distribution of the samples in the majority class is stationary (the samples are very close to each other) undersampling may work very well as we don't loose lot of information by deleting some samples.

III. Our Approach

In this section, we present our **MINN-AE** model and the other classifiers we tested. We compared MINN-AE to baseline artificial neural networks and state-of-the-art classifiers such as logistic regression, random forest and gradient boosting.

A. State-of-the-art Classifiers

We compared the artificial neural network models to three state-of-the-art classifiers: logistic regression (LR), random forest (RFC) and gradient boosting (GBC). We chose these three classifiers because they are commonly used and provide reasonably good performance on tabular data. We compare their performance to those of of artificial neural networks based classifiers. The optimal hyper parameters are chosen using a grid search.

B. Baseline Artificial Neural Network

We first tested some baseline Multi-Layer Perceptrons (MLP) models consisting of a single input layer, multiple hidden layers, and an output layer. These models take an invoice as input and predicts if it's fraud or not. The number of layers and the number of neurons in each layer are variables (hyperparameters) that must be chosen carefully for neural network models to give good results. We refer to the baseline neural network as **BNN**. The BNN is a simple multilayers perceptron model where all the features are concatenated and feed to the model. We tested some version of BNN using the loss functions as describe in subsection IV-B. **BNN weighted** stands for BNN with weighted loss function, **BNN focal** with focal loss function, **BNN mfe** with the mean false error loss function and **BNN rus** the best BNN obtained by random under sampling.

C. MINN-AE Model's Architecture

MINN-AE is made up of two different inputs layers. The first input layer receives the data related to the claims in case of medicare data or the features related to the transaction in case of transaction fraud detection. The second input layer receives the features related to the healthcare provider in case of medicare fraud detection or the credit card holder (or the receiver) in case of transaction fraud detection. The model is thus composed of two blocks which meet at the end. Each block consists of an input layer, hidden layers and an output layer. The outputs of the two blocks are then concatenated to form a single vector which is feed to a fully connected layers. Such an architecture makes it possible to use simultaneously many source of data without mixing them. In our version of the multi-input model, the second block is an autoencoder. We first trained the autoencoder on the provider level (or credit card holder) features. This autoencoder learns to reconstitute the provider behavior over time. Then we used the latent vector from the autoencoder as an input vector for our final model. In this architecture, the autoencoder plays a dimension reduction role for the provider data and its latent vector describes the provider behavior. Note that the autoencoder parameters remain constant when learning the final model. The model's architecture is presented in Fig. 1.

IV. Experimental Design

We applied our model architecture to several state-of-the-art techniques to deal with class imbalance such as random under sampling, weighted loss, focal loss etc.

A. Resampling Methods

We compare the MINN-AE performances to those of the state-of-the-art classifiers, in the under sampling scenario. We choose random undersampling first because according to the literature it tends to give better performance than over sampling. Moreover, as all our data sets are huge, undersampling is easier to performance than oversampling. For each pair (dataset, model), we use a subsample (10%) of the training data set to chose the best undersapling rate $r \in (0, 1)$. r is the ratio between the number of samples in the minority class and that of the majority class, if $r = 1$, the classes are balanced.

B. Algorithm Level Methods

In this subsection, we describe the algorithm level methods tested with our classifiers.

1) Weighted Cross Entropy: This cost function integrates class-wise weights. The loss of each data is multiplied by the weight of the class it belongs to. The total cost function is written as follows:

$$Weighted\ CE\ loss = -\sum_{i=1}^{C} w_i P_i \log(P_i)$$

With w_i the weight associated to class i, P_i the probability of class i and C the total number of classes.

2) Focal Loss: The Focal loss function is written as follows:

$$FL(p) = \alpha(1-p)^{\gamma} \log p$$

For easy classified samples ($p- > 1$) the modulation factor tends towards 0 which reduces their importance in the final loss function. Moreover, if a sample is badly classified ($p- > 0$), the modulation factor is close to 1 and the cost function is little affected. The parameter γ controls the contribution of a sample in the final loss according to its classification complexity.

3) MFE Loss: The mean false error (MFE) cost function is written as a weighted average of the errors of the two classes. The final loss function is a sum of to means: mean false positive error (FPE) and mean false negative error (FNE). **MFE** $= FPE + FNE$ and **MSFE** $= FPE^2 + FNE^2$

$$FPE = \frac{1}{N}\sum_{i=1}^{N}\sum_{n}\frac{1}{2}(d_n^{(i)} - y_n^{(i)})^2 \qquad FPE = \frac{1}{P}\sum_{i=1}^{P}\sum_{n}\frac{1}{2}(d_n^{(i)} - y_n^{(i)})^2$$

N is the number of negative samples, P the number of positive samples, $d^{(i)}$ the true label of sample i, $y^{(i)}$ the predicted label for sample i.

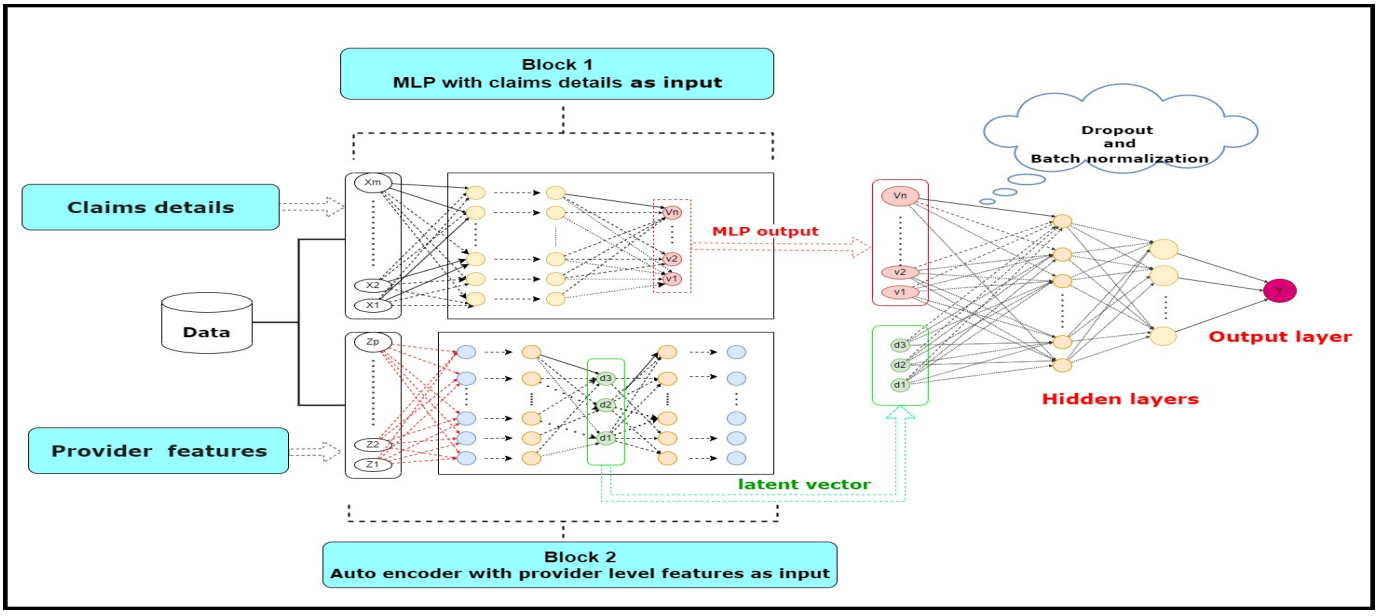

Fig. 1: Visualization of the proposed neural network architecture. Block 1 receives features related to the invoice. Block 2 receives features related to the provider behavior and trains an autoencoder. The latent vector of the autoencoder and the output of block 1 are concatenated and used as input for the next hidden layers of the model.

C. Hyperparameters Optimization

The dataset has been separated into a training dataset (80 %) and test (20 %) dataset. To avoid any risk of data leakage, we split the dataset according to the healthcare provider Id. The models are trained using a k-folds cross-validation ($k = 10$ in our case). The final performance is computed on the test set. The final value of each performance metric is computed by taking the average of the ten measurements obtained during the ten iterations of the cross-validation (see the supplementary materials for more details).

D. Choice of the Optimal Decision Threshold

To improve overall performance and better illustrate the efficacy of our model in detecting fraud, we apply moving thresh-old to each classifier independently. The choice of the optimal decision threshold is made on a subset of the validation set during the training phase. Thus during each iteration, we choose the optimal threshold by varying it between 0 and 1. We choose the threshold that maximizes the AUC(ROC) score.The threshold selection algorithm is described in more details in the supplementary materials.

E. Performance Metrics

The classifiers are evaluated using the AUC (ROC) and the area under the curve of precision-recall curve, denoted as AUC (PRC). We chose to use AUC (PCR) metric in addition to the AUC (ROC) because as Saito et al. [10] show in their study, the AUC (ROC) may not be well suited in case of highly imbalanced classes. In their article [10], these authors showed that AUC (ROC) could be misleading when applied in imbalanced classification scenarios instead AUC (PRC) should be used. Their study showed via multiple simulations that AUC (ROC) fails to capture the variation in class distribution contrary to the AUC (PRC).

V. Experimental Data sets

In order to evaluate our method's capabilities, we compared its performance to those of four state-of-the-art classifiers on four other publicly available bench mark data sets. The data sets are described in details in the supplementary materials.

The Centers for Medicare and Medicaid Services (CMS) publishes a series of publicly available data each year containing information on the use and payments of medical procedures, services and prescription drugs provided to beneficiaries as well as data on physicians and other actors in the healthcare system [11]. In this study we used Part B and Part D of the Medicare Provider Utilization and Payment Data of the CMS data sets from the period 2017-2019. The initial data set contains 14.668.478 rows and 44 features. These **CMS data** were combined with the Office of Inspector General's list of excluded individuals and entities (LEIE) [12] containing the list of healthcare providers excluded from the healthcare system for illegal activity. We created some additional features for the providers by aggregating the variables at the invoice level. For each provider we created new variables by taking the mean, the variance, the sum, the skewness coefficient of the numerical variables per trimester. After cleaning and preprocessing, the final data set has a fraud rate of 0.05%.

The **Kaggle medicare** data set is available on kaggle [13] and contains three kind of information like the CMS data set: hospitalization requests (Inpatient Data), outpatient care requests (Outpatient Data) and beneficiary information

(Beneficiary Details Data). These three tables have been combined into a single table containing patient information as well as invoices. The **Electricity Consumption Fraud Detection** data set is available on kaggle. It comes from a real-world electricity consumption. The goal is to detect fraudulent transactions. **Online Payments Fraud Detection Data set** is available on kaggle for fraud detection modeling, testing and debugging purposes. It contains 6362620 rows of transactions.The data set is highly imbalanced with an imbalance rate of 0.13%. **Credit Card Transactions Fraud Detection Data set** is a simulated credit card transaction containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020. It It covers credit cards of around 1000 customers doing transactions with a pool of 800 merchants. The imbalance rate is 0.57%.

VI. Results and discussions

In this section we discus the performance of the classifiers on our experimental data sets.

A. Results on CMS Medicare Data Set

In this subsection we present the results of our classifiers on the CMS medicare data sets from 2017 to 2019 and the kaggle medicare data set.

From Table I, we can see that despite the class imbalance in the training data, MINN-AE outperforms all other classifiers on the CMS and kaggle medicare data sets. On the CMS data set, MINN-AE has the best AUC(ROC) (0.796) and the best AUC (PRC) (0.215). On the kaggle data set, it has best AUC (ROC) (0.794) and the best AUC (PRC) (0.765). It is followed by the multi-layers perceptron models (BNN). These results suggest that without any re-balancing technique, artificial neural network based classifiers perform better than state-of-art classifiers on highly imbalanced medicare data set.

In the random undersampling scenario (see Table II), BNN has the highest performance in terms of AUC (ROC)=0.795) on the CMS data set. But when we use the AUC (PCR) as performance metric, Gradient boosting has the best results (AUC=0.204). On the kaggle medicare data set, MINN-AE has the highest performance (ROC (PCR)=0.729 and ROC (ROC)=0.855). Our experiments show that under sampling more the majority class ($r > 0.3$) leads to decrease in the model's performance. This indicates that, when $r > 0.3$, we loose valuable information necessary to the learning process. Indeed, the CMS data set exhibits both big data and class rarity, and under sampling leads to the suppression of millions of negative samples. When we apply algorithm level methods (see table III) to deal with class imbalance, MINN-AE with weighted loss has the highiest AUC (ROC) (0.785). MINN-AE combined with focal loss has the best AUC (PCR) (0.259).

The advantage of MINN-AE is that the autoencoder separates the providers into homogeneous groups and creates contextual features. In fraud detection the context matters. For example two providers can provide very similar claims but due to their previous behaviors (context) one will be considered fraudulent and the other one genius. State-of-the-art classifiers (logistic regression, random forest, Gradient boosting) perform worst than artificial neural network classifiers because they fails to capture complex structures in sequence data sets and large scale data. Deep learning models have excellent capabilities in learning expressive representations of complex data such as high-dimensional data, temporal data and spatial data.

TABLE I: Performance on experimental data sets without class balancing. Mean time refers to the execution time expressed in minutes (lower the best).

Classifier	Metric	CMS 2017-2019	kaggle medicare	Electricity	Payment	Credit Card
No skill	*AUC(ROC)*	0.5	0.5	0.5	0.5	0.5
	AUC (PRC)	0.01	0.03	0.056	0.0013	0.0058
	Mean Time	-	-	-	-	-
LR	AUC(ROC)	0.761	0.674	0.547	0.997	0.847
	AUC (PRC)	0.133	0.636	0.109	0.993	0.139
	Mean Time	0.9	0.32	2.54	1.29	0.58
RFC	AUC(ROC)	0.771	0.714	0.574	0.998	**0.969**
	AUC (PRC)	0.177	0.643	0.125	0.997	**0.837**
	Mean Time	0.39	0.37	4.84	37.59	13.65
GBC	AUC(ROC)	0.739	0.715	0.574	**0.998**	0.857
	AUC (PRC)	0.145	0.617	0.125	**0.997**	0.857
	Mean Time	2.37	15.43	14.93	16.48	47.03
BNN	AUC(ROC)	0.765	0.863	0.566	0.998	0.942
	AUC (PRC)	0.166	0.739	0.121	0.997	0.708
	Mean Time	7.1	2.54	8.90	6.05	4.16
MINN-AE (Ours)	AUC(ROC)	**0.796**	**0.794**	**0.627**	0.997	**0.943**
	AUC (PRC)	**0.215**	**0.765**	**0.158**	0.995	**0.773**
	Mean Time	8.7	8.65	8.94	7.69	3.95

TABLE II: Performances with resampling methods for addressing class imbalance.

Classifier	Metric	CMS 2017-2019	kaggle medicare	Electricity	Payment	Credit Card
LR rus	AUC(ROC)	0.764	0.830	0.547	0.997	0.847
	AUC (PRC)	0.099	0.661	0.109	0.984	0.138
	Mean Time	0.1	0.22	0.93	0.62	0.45
RFC rus	AUC(ROC)	0.773	0.849	0.574	0.998	0.970
	AUC (PRC)	0.183	0.695	0.126	0.996	0.832
	Mean Time	0.13	0.83	2.0	1.22	0.97
GBC rus	AUC(ROC)	0.763	0.845	0.574	**0.998**	**0.978**
	AUC (PRC)	**0.204**	0.681	0.125	**0.997**	**0.877**
	Mean Time	0.55	13.44	6.25	0.76	1.86
BNN rus	AUC(ROC)	**0.795**	0.720	0.564	0.997	0.928
	AUC (PRC)	0.194	0.512	0.119	0.995	0.542
	Mean Time	1.60	1.20	2.74	2.36	1.35
MINN-AE rus (Ours)	AUC(ROC)	0.761	**0.855**	**0.626**	0.653	0.593
	AUC (PRC)	0.165	**0.729**	**0.161**	0.020	0.014
	Mean Time	4.61	6.36	8.02	2.15	1.07

TABLE III: Performance with algorithm level methods for addressing class imbalance.

Classifier	Metric	CMS 2017-2019	kaggle medicare	Electricity	Payment	Credit Card
LR weighted	AUC(ROC)	0.765	0.827	0.548	0.997	**0.972**
	AUC (PRC)	0.104	0.620	0.109	0.993	**0.857**
	Mean Time	0.10	0.51	6.85	1.29	0.58
RFC weighted	AUC(ROC)	0.770	0.807	0.571	0.998	0.962
	AUC (PRC)	0.154	0.658	0.124	0.997	0.837
	Mean Time	0.29	2.35	4.39	37.59	0.97
BNN weighted	AUC(ROC)	0.766	0.860	0.569	0.998	0.949
	AUC (PRC)	0.147	0.733	0.122	0.995	0.519
	Mean Time	1.42	2.96	8.47	6.53	3.98
BNN focal	AUC(ROC)	0.774	0.861	0.568	**0.999**	0.950
	AUC (PRC)	0.158	0.737	0.122	**0.997**	0.729
	Mean Time	1.53	2.50	9.17	7.31	4.27
BNN mfe	AUC(ROC)	0.753	**0.864**	0.563	0.996	0.862
	AUC (PRC)	0.115	**0.741**	0.120	0.990	0.365
	Mean Time	1.16	6.03	9.24	6.84	4.22
MINN-AE weighted	AUC(ROC)	**0.785**	0.856	0.623	0.997	0.949
	AUC (PRC)	0.208	0.718	0.154	0.986	0.539
	Mean Time	2.52	8.45	9.14	7.47	4.26
MINN-AE focal	AUC(ROC)	0.764	0.850	0.549	0.897	0.765
	AUC (PRC)	**0.259**	0.725	**0.383**	0.895	0.464
	Mean Time	3.87	7.36	9.07	8.26	3.98

B. Results on Bench Mark Data Sets

In this subsection, we compare the performance of the classifiers with fraud detection data sets from credit card payment and online transaction. Tables I II III show that, MINN-AE gives good results depending on the size of the data seet.

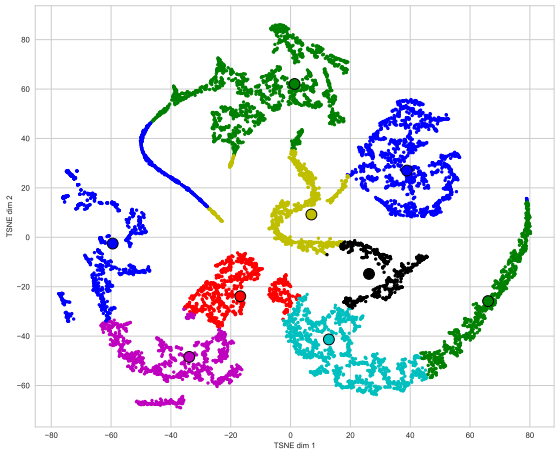

Fig. 2: Mean-shift clustering on the autoencoder latent vector. This output of the autoencoder separates the providers into homogeneous groups.

On the Electricity and the Credit card data sets, MINN-AE outperforms all the other classifiers in terms of AUC (ROC) and AUC (PCR) (see Table I). On the Online payment data set, MINN-AE is the second best classifier (AUC(PRC)=0.995) behind Gradient boosting(AUC(PRC)=0.997). Note that state-of-the-art classifiers (random forest, logistic regression and gradient boosting) performe very well on the **Online payment** and the **Credit card** data sets despite the high imbalance rate but fail when applied to the **CMS** and the **Electricity** data sets. One explanation is that the last two data sets are more complex than the other ones. Therefore, is it is more difficult to separate fraudulent transactions from legitimate ones in the **CMS** data set than in the other data sets.

C. Discussions

Our results suggest that, in medicare fraud detection framework, using an autoencoder to embed the provider level features makes it easier for the neural network to separate fraudulent transaction from legitimate ones. The autoencoder acts like a dimensional reduction layer and also learns the provider behavior. The model is also robust toward the imbalance class due to the fact that the latent features extracted by the autoencoder have strong clustering power. The latent features allows the model to group the providers into into homogeneous groups (see Fig. 2) and makes it easier to identify fraudulent behaviors. The experiments also suggest that this kind of architecture works better when we have enough provider level features to train the autoencoder part. Table II shows that the MINN-AE architecture does not work very well in under sampling scenario due to the fact that the model has lot of parameters to train. In addition, we found that maintaining a sufficient representation of the majority class may be more important than reducing the level of class imbalance. These results are in agreement with those of Johnson et al. [4] who used the CMS data over the period 2012–2016.

Our experiments also show that state-of-the-art classifiers like logistic regression, random forest and gradient boosting can outperform neural network based classifiers on tabular data sets depending on the complexity of the data. It is therefore important to test simple classification models before very complex and expensive neural network models.

VII. Conclusion

In this study, we proposed a deep neural networks with multiple inputs called **MINN-AE** to detect medicare frauds. Our model has an autoencoder component that learns contextual features from the input data. The results showed that this kind of architecture outperforms a classical multi-layer perceptron models using a single input layer. The model is also robust toward the imbalance class issue. The results also suggest that employing MINN-AE models with data sampling techniques or algorithm level methods for addressing class imbalance can improve the model's performance. However, when undersampling, maintaining a sufficient representation of the majority class may be more important than reducing the level of class imbalance.

References

[1] R. A. Bauder and T. M. Khoshgoftaar, "The detection of medicare fraud using machine learning methods with excluded provider labels," in *The Thirty-First International Flairs Conference*, 2018.

[2] Q. Liu and M. Vasarhelyi, "Healthcare fraud detection: A survey and a clustering model incorporating geo-location information," in *29th world continuous auditing and reporting symposium (29WCARS), Brisbane, Australia*, 2013.

[3] M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, "Big data fraud detection using multiple medicare data sources," *Journal of Big Data*, vol. 5, no. 1, p. 29, 2018.

[4] J. M. Johnson and T. M. Khoshgoftaar, "Medicare fraud detection using neural networks," *Journal of Big Data*, vol. 6, no. 1, pp. 1–35, 2019.

[5] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 935–942.

[6] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *2016 international joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 4368–4374.

[7] H. Wang, Z. Cui, Y. Chen, M. Avidan, A. B. Abdallah, and A. Kronzer, "Predicting hospital readmission via cost-sensitive deep learning," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 15, no. 6, pp. 1968–1978, 2018.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[9] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, 2019.

[10] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[11] US and Government. (2018) Cms. CMS. [Online]. Available: https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/medicare-provider-charge-data/physician-and-other-supplier

[12] O. of Inspector General. (2020) List of excluded individuals and entities. OIG. Exclusion authorities. [Online]. Available: https://oig.hhs.gov/exclusions/authorities.asp

[13] Healthcare provider fraud detection analysis. Kaggle. [Online]. Available: https://www.kaggle.com/rohitrox/healthcare-provider-fraud-detection-analysis/metadata