Machine Learning for anomaly detection. Performance study considering anomaly distribution in an imbalanced dataset

1st Salma El Hajjami
IASSE Laboratory
ENSA, USMBA
Fez, Morocco
salma.elhajjami@usmba.ac.ma

2nd Jamal Malki
L3i Laboratory
La Rochelle University
La Rochelle, France
jmalki@univ-lr.fr

3rd Mohammed Berrada
IASSE Laboratory
ENSA, USMBA
Fez, Morocco
mohammed.berrada@gmail.com

4th Bouziane Fourka
aYaline R&D Laboratory
aYaline
Poitiers-Futuroscope, France
bfourka@ayaline.com

***Abstract*—The continuous dematerialization of real-world data greatly contributes to the increase in the volume of data exchanged. In this case, anomaly detection is increasingly becoming an important task of data analysis in order to detect abnormal data, which is of particular interest and may require action. Recent advances in artificial intelligence approaches, such as machine learning, are making an important breakthrough in this area. Typically, these techniques have been designed for balanced data sets or that have certain assumptions about the distribution of data. However, the real applications are rather confronted with an imbalanced data distribution, where normal data are present in large quantities and abnormal cases are generally very few. This makes anomaly detection similar to looking for the needle in a haystack. In this article, we develop an experimental setup for comparative analysis of two types of machine learning techniques in their application to anomaly detection systems. We study their performance taking into account anomaly distribution in an imbalanced dataset.**

Keywords: Anomaly Detection, Data Analysis, Artificial Intelligence, Machine Learning, Imbalanced Data.

I. Introduction

In general, anomaly detection consists in detecting rare events or data which are significantly different from normal. Study anomaly detection is important because abnormal elements carry interesting data that can be used in a wide variety of application areas. Among examples of common applications, in bank transaction data, an anomaly means that there may have been a fraudulent transaction [6], [15]. In computer security, anomaly detection can be used to monitor network traffic and identify intrusions [4]. Anomalies in medical data can be monitored to provide preventive health warnings [7].

Recently, several approaches for anomaly detection have been developed, including artificial intelligence based approaches using machine learning algorithms. The detection of anomalies in machine learning can be carried out in two paradigms which strongly depend on the availability of labels. For events where each observation is associated with a label (normal or abnormal), the anomaly detection is said to be supervised. In this mode, the objective is to train a model to better separate normal data from abnormal. The second paradigm is unsupervised, in which no information is available a priori. The approach consists in building a normality model based solely on proximity or density assumptions. For example, an event is considered abnormal and is supposed to be different from the others, if it is far from its closest neighborhood or if it is in regions with low density in the description space [3].

Traditionally, most anomaly detection systems have been designed for balanced datasets or that have certain assumptions about the distribution of data. However, real applications are more often faced with imbalanced data distributions. A data set is called imbalanced when one class is underrepresented (minority class) compared to another (majority class). Having few instances of a class means that the learning algorithm will often be unable to generalize the behavior of the minority class, which will impact its final performance [5].

The minority class's prediction accuracy is of crucial importance because this class is generally of great interest as in the cases of anomalies. Thus, a bad prediction of the minority class has a much higher cost compared to a bad prediction of a majority class instance. In this context, we are interested in machine learning approaches applied to anomaly detection occurring in financial data transactions. In our study, we use "Credit card fraud detection" dataset provided by Kaggle [9]. The particularity of the problem is that the anomalies constitute the minority class are represented by a very small percentage of the dataset. To study this problem in this work, we consider an experimental configuration based on machine learning algorithms within the framework of Apache Spark's Machine Learning library. The main objective of this work is to study the performance of supervised and unsupervised machine learning techniques for anomaly detection, considering anomaly distribution in an imbalanced dataset. To this end, we are developing an experimental setup in which these techniques can be compared equally. The performances of the two groups of methods are evaluated based on Accuracy and Area Under ROC Curve (AuRoc) metrics.

PLAIBDE Project. Programme Opérationnel FEDER-FSE de la région Nouvelle Aquitaine - France

The rest of the paper is organized as follows. Section II is devoted to the state of the art. We present an overview of anomaly detection problem and discuss recent work related to machine learning for anomaly detection. In section III, we present supervised and unsupervised machine learning methods used in the field of anomaly detection and considered in this work. In section IV, we present our experimental setup. In section V the experimental results are presented and discussed. Lastly, section VI presents the conclusion and perspectives.

II. Background and related work

In this section, we present an overview of anomaly detection problem and discuss recent works related to machine learning for anomaly detection.

A. Anomaly detection

Anomaly detection is a data analysis process that detects abnormal or aberrant data in a given dataset [3]. This is an interesting area of data analysis research and refers to the process of finding data models that do not conform to the expected behavior. These nonconforming models are often referred to as anomalies, but depending on the application domain, we can find other descriptions such as outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants. It has been widely studied in statistics and machine learning, and also been described as synonymous with outlier detection, novelty detection, deviation detection and exception mining. Although researchers define an anomaly in different ways, there is a widely accepted definition introduced by Hawkins D.M. [8]: "An anomaly is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism".

Anomaly detection is considered crucial because it indicates important but rare events and may prompt critical actions. It has been widely applied in countless application domain such as biomedical, financial, video surveillance, security, network systems and fraud detection. For example, abnormal data in credit card transactions could indicate fraudulent activities. An unusual pattern of a network traffic may mean that a computer has been hacked and data is being transmitted to unauthorized destinations. An anomaly in an MRI (magnetic resonance imaging) image could indicate the presence of a malignant tumor.

B. Machine learning for anomaly detection

Different machine learning methods are proposed for the anomaly detection problems. They can be broadly classified into two general approaches: supervised and unsupervised.

1) Supervised approach for anomaly detection: Supervised approaches are commonly based on classification methods. They require a dataset where the data are labeled normal or abnormal to build the predictive model. A typical approach in such cases is to construct a predictive model for normal classes and anomaly classes. New data are compared to this models to determine their classes. The supervised anomaly detection raises two major problems. First, the abnormal cases are much less numerous than the normal cases in the learning data. Secondly, this technique is hardly very relevant because the anomalies are known and correctly labeled. For many applications, the anomalies are not known in advance or may appear spontaneously as novelties during the test phase [3][11]. The most common supervised algorithms are: Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, Gradient Boosted Trees [11][2].

2) Unsupervised approach for anomaly detection: Unsupervised approaches consider a set of unlabeled data. Also, there is no distinction between a set of learning and test data. As an alternative, unsupervised approaches are based on two basic assumptions. First, they assume that most data is normal and only a very small piece of the data is abnormal. Second, they predict that any anomaly is statistically different from normal samples. According to these two hypotheses, groups of data of similar instances that appear frequently are assumed to be normal data, while instances that are very different from the majority are considered as anomalies [3], [11]. The most common unsupervised algorithms are: Isolation Forest, Gaussian Mixtures Model and K-Means [11], [2].

C. Machine learning for anomaly detection over imbalanced data

Traditionally, most anomaly detection systems have been designed for balanced data sets or that have certain assumptions about data distribution, such that data should have a predetermined and fixed distribution. However, real applications are different from this constrained situation. In fact, huge applications, namely credit card fraud detection, climate monitoring, network intrusion detection, etc., are rather confronted with imbalanced data distribution. It is a difficult situation that lead to inaccurate results and making it an area of interest to researchers.

In this regard, various approaches have been proposed to deal with imbalanced datasets issue and improve the performance of machine learning approaches. These last could be mainly divided into two categories. The first category of methods considers the problem at the data level, using data resample techniques (undersampling, oversampling). The second type of approaches considers the problem at the algorithm level.

In [1], authors investigated the performance of Naïve Bayes, K-Nearest Neighbor and Logistic Regression on highly skewed credit card fraud data. They applied a hybrid technique of under-sampling and oversampling to handle the skewed data. The performance of these techniques is evaluated based on accuracy, sensitivity, specificity, precision, Matthews correlation coefficient and balanced classification rate. Their results show that K-Nearest Neighbour performs better than Naïve Bayes and Logistic Regression algorithms. Autors of [12] proposed a comparative performance of ten different machine learning algorithms that have been classified into two groups namely classification algorithms and ensemble learning group. The comparative study considers a credit card fraud imbalanced dataset prepared by using under-sampling method. Two ensemble learning algorithms group have been found to perform better when the used dataset does not include the "Time" feature. However, for the classification algorithms group, three classifiers are found to show better predictive accuracies when all attributes are included in the used dataset.

In [6], authors used many supervised machine learning algorithms to detect credit card fraudulent transactions using a real-world imbalanced dataset. Authors apply an under-sampling technique to balance the data, and identify the most important variables that may lead to higher accuracy in credit card fraudulent transaction detection. Their results show that stacking classifier which is used Logistic Regression as meta classifier is most promising for predicting fraud transaction in the dataset, followed by the Random Forest and eXtreme Gradient Boosting classifier. In [10], authors compared several machine learning techniques and investigated their suitability as a "scalable algorithm" when working with highly imbalanced massive or "Big" datasets. The experiments were conducted on two highly imbalanced datasets using Random Forest, Balanced Bagging Ensemble, and Gaussian Naïve Bayes. Then, they applied various balancing techniques such as Random Under Sampling, Random Over Sampling, various flavours of SMOTE (original, borderline1, borderline2, SVM), SMOTEENN, and SMOTETomek to both datasets. They observed that many detection algorithms performed well with medium-sized dataset but struggled to maintain similar predictions when it is massive. Random Forest with Random Under Sampling is proven to be scalable and capable of fraud detection with highly imbalanced massive datasets.

Autors of [13] evaluate popular methods of oversampling minority class examples and undersampling majority class examples for their capability of improving imbalanced ratio of five highly imbalanced datasets from different application domains. Authors study the effect of balancing on classification results. They observed that adaptive synthetic oversampling approach can best improve imbalanced ratio as well as classification results. However, undersampling approaches offer better overall performance on all datasets. Autors of [4] adopted deep variational autoencoders to generate new data and adjust data imbalance into more favorable balanced data. From the results, they observed that the resulting balanced data can practically lead to better classification accuracy. Indeed, when facing unknown attacks and in order to solve the over-adaptation problem in intrusion detection models formation, this can ensure that the trained intrusion detection model will not misjudge new types even if they are not in the training dataset. In [15], authors designs a credit card fraud prediction model based on cluster analysis and integrated support vector machine. They adjust and reduce imbalanced state based on K-Means clustering analysis combined with more than half of the random samples.

In [14], authors compared certain machine learning algorithms for detection of fraudulent transactions such as Logistic Regression, Random Forest, Naïve Bayes and Multilayer Perceptron. Because the dataset was highly imbalanced, SMOTE technique was used for oversampling. They established that Random Forest algorithm gives the best results (i.e. best classifies whether transactions are fraud or not). This was established using different metrics, such as recall, accuracy and precision. For this kind of problem, this work shows that it is important to have recall with high value. Feature selection and balancing of the dataset have shown to be extremely important in achieving significant results.

III. Supervised and unsupervised machine learning for anomaly detection

In this section, we present supervised and unsupervised machine learning methods used in the anomaly detection field and considered in this work. More precisely, we are interested by methods having an official implementation within the open source distributed computing framework Apache Spark. The whole system will be described in the section presenting the system architecture.

A. Machine learning methods studied

1) Supervised methods:

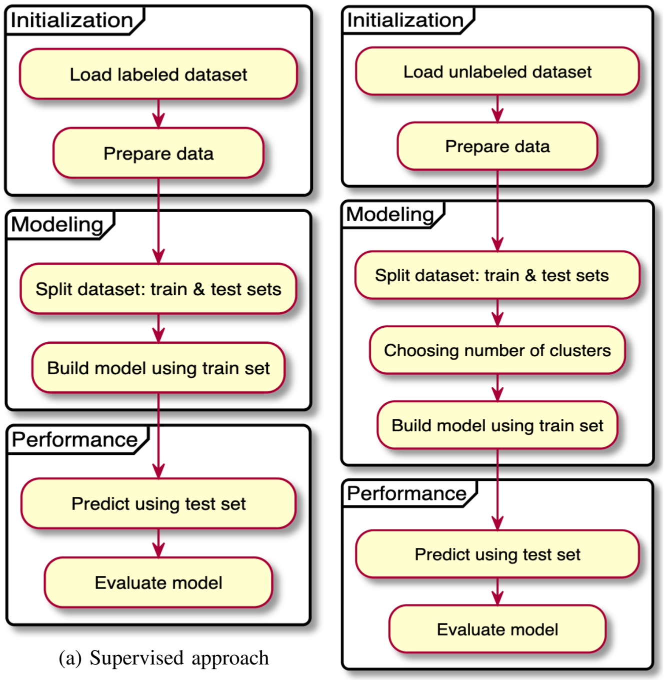
- Logistic Regression (LR): uses a functional approach to estimate the probability of a binary response based on one or more variables (features). Normally, it is used when there are only two results: the event occurs or does not occur.
- Decision tree (DT): utilizes a top-down approach in which the root node creates binary splits until a certain criteria is met. This binary splitting of nodes provides a predicted value based on the interior nodes leading to the terminal (final) nodes.
- Random Forest (RF): used for classification and regression. It is a forest because it has a group of decision trees that each result gives a planned or suggested outcome. It is random because it randomly selects part of the data set to be formed. Once each decision tree has made a decision, the forest will make a prediction based on the majority of the votes of the trees.
- Gradient-Boosted Tree (GBT): like (RF) because they are both a set of decision trees that make predictions. The difference is that in Random Forest, the prediction of a tree is independent of that of other trees. With gradient boosted trees, the trees are iteratively driven. When a tree makes its prediction, the next tree uses this previous prediction to make its own.
- Naïve Bayes (NB): used to calculate the set of probabilities by counting the value and frequency of values in a given set of data. It is based on Bayesian Theorem with the assumption that each feature is independent.
- Support Vector Machine (SVM): attempts to define a separation line that separates the data points lying in the different classes, called a hyperplane, so that when a new sample comes in, it is classified based on which side of the gap they fall in.
- MultiLayer Perceptron (MLP): a classifier based on the feedforward artificial neural network. MLP maps a list

of input nodes to a list of output nodes. These two sets of nodes are in two different layers. To move from one layer to another, each node applies a calculation based on input and weighting. There may be many layers of nodes hidden between the input and output nodes that increases the accuracy of the map but takes longer to train and predict.

2) Unsupervised methods:

- K-Means (KM): a clustering algorithm that groups data points into a predefined number of clusters. After a random initial assignment of examples to k clusters, the cluster centers are calculated and the examples are assigned to clusters whose centers are closest. The process is repeated until the cluster centers do not change significantly. Once the cluster assignment is fixed, the average distance from an example to cluster centers is used as the score.
- Bisecting K-Mean (BKM): a hierarchical clustering using a division approach (top-down). All observations start in a cluster and divisions are done recursively as one goes down in the hierarchy.
- Gaussian Mixture Model (GMM): a probabilistic learning model where each input sets is modeled by itself, without comparison with other groups. It also tries to construct the best possible probability distribution for each group using a set of Gaussian probability distribution functions called Gaussian mixtures.

B. General approach



(a) Supervised approach

(b) Unsupervised approach

Fig. 1: General approach considered

The figure 1 shows the general approach considered in this work. Both supervised and unsupervised approaches are based on the same master components: Initialization, Modeling and Performance.

1) Initialization: Both approaches load a dataset and prepare the contained data in order to build the machine learning models. Preparation process transforms the available data into a data that can be used to train and evaluate machine learning models. It consists in cleaning, transforming (standardization, normalization, etc.) to ensure uniformity. The prepare process can also include selecting the relevant features. Supervised approach takes labeled dataset while it's unlabeled for the unsupervised case.

2) Modeling: The modeling process divides the loaded dataset into training and test sets. This process is carried out by training a given algorithm on a the training dataset associated with their labels, for supervised case. For an unsupervised algorithm, prior knowledge of labels is not essential, the model is built with the appropriate number of clusters.

3) Performance: Predictions obtained from the test set can be used to determine the performance of the trained model.

IV. Experimental Setup

In this section we describe the different parts of our experimental setup for anomaly detection.

A. Dataset description

For this work, we use the credit card fraud detection dataset from Kaggle [9]. The dataset contains transactions made by credit cards in two days of September 2013 by European card holders. Table I gives dataset's statistics and shows that the positive class, which means frauds, account for 0.172% of all transactions. Therefore, this dataset is highly imbalanced [5].

TABLE I: Kaggle credit card fraud dataset details

Transactions	Negative class	Positive class	Columns
284 807	284 315	492	31

Each line of the dataset can be represented as:

$$V^i = \left(Time^i, V_1^i, V_2^i, \ldots, V_{28}^i, Amount^i, Class^i\right) \qquad (1)$$

In equation 1, the values from V_1^i to V_{28}^i are the principal components obtained with a PCA transformation. They are numeric. The only components which have not been transformed with PCA are $Time^i$, and $Amount^i$:

- $Time^i$: contains the seconds elapsed between each transaction and the first transaction in the dataset ;
- $Amount^i$: is the transaction amount ;
- $Class^i$: is the response component and it takes value 1 in case of fraud and 0 otherwise.

B. Training and test datasets

Machine learning methods split input dataset into training and test sets. Generally, if the data are not correlated, then the training and test sets can be obtained randomly. But in our case, we know the time elapsed between transactions. Therefore, the data are time correlated. Thus, we split the dataset according to the time of occurrence. As there is no rule-of-thumb for how to divide a dataset into training and test sets, we consider the two important ratio used in machine learning: 70/30 and 80/20. Table II shows that in the case of 80/20 rule, the percent of positive class of test dataset falls below 20% of the total fraud. So, this can negatively impact the performance of the methods studied. In conclusion, we choose the 70/30 rule because it gives a percent of positive class close to 20%.

TABLE II: Training and test dataset statistics

		Total	**Positive class**
	Dataset	284 807 – 100%	492 – 100%
Rule 70/30	**Training dataset**	199 364 – 70%	384 – 78%
	Test dataset	85 443 – 30%	108 – **22%**
Rule 80/20	**Training dataset**	227 845 – 80%	417 – 85%
	Test dataset	56 962 – 20%	75 – **15%**

C. Spark machine learning platform

Apache Spark is a processing engine that provides both real time, batch and streaming processing of data. Spark's framework is designed for data science and its abstraction makes data analysis easier. Spark has the ability to cache the dataset in memory, speeds up the iterative data processing thus, making it an ideal processing engine, especially for machine learning. Spark is based on Scala, but offers APIs for Java, Python and R programming languages. For our experiments, we use the Scala implementation.

As pointed before, this work is based on the Spark machine learning library "MLlib". This library offers common learning algorithms such as classification, regression, clustering, and collaborative filtering. The primary Machine Learning API for Spark is now the DataFrame-based API in the *spark.ml* package witch is an evolution of the RDD-based API. As per Apache Spark documentation: "Spark can run both by itself, or over several existing cluster managers". In our work, we choose the Hadoop Yarn cluster manager for Spark deployment (Fig. 2).

There are two deploy modes to launch Spark applications on Yarn: cluster mode and client mode. The Fig. 2 shows the adopted architecture based on the cluster mode where everything runs inside the cluster. Indeed, the Spark driver runs inside an application master process which is managed by Yarn on the cluster. In this mode the client can go away after initiating the application. In client mode, the driver runs in the client process, and the application master is only used for requesting resources from Yarn. In our experiments, we will run applications based on algorithms which can take a long time processing. For this reason, cluster mode is more appropriate. We think that client mode is well suited for interactive jobs, but applications will fail if the client stops. Yarn cluster mode needs an appropriate memory allocation configuration.

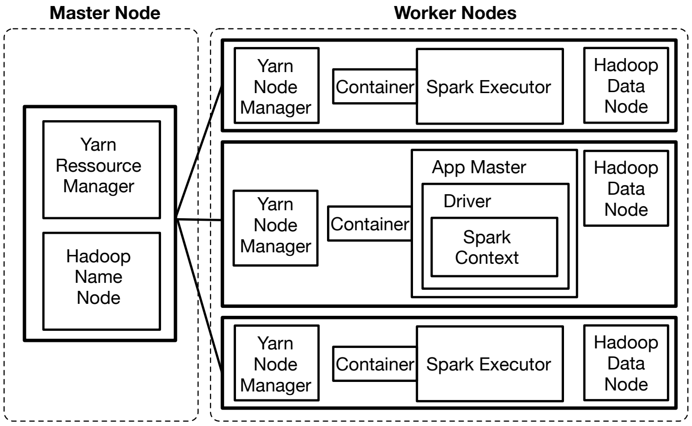


Fig. 2: Hadoop Yarn cluster mode for Spark

As shown in the Fig. 2, our experimental plateform contains one master node and three worker nodes configured as follows:

- Operation system: ubuntu-16.04.3-server-amd64
- Hard disk: 500GB SSD type
- Processors: 16GB RAM (4 sockets, 4 cores)
- Network: 2x10GB SFP+

V. Evaluation and Results

A. Performance Metrics

Spark "MLlib" library provides a series of metrics to assess the prediction of the resulting models. In our case, the major challenge is to tackle the imbalance problem, since legitimate transactions are much more numerous than fraudulent transactions (less than 1% of total transactions). This problem often leads to extremely high accuracy (Definition 2) where a model can reach up to 99% of the prediction accuracy, ignoring the 1% of minority class cases. In other words, accuracy does not reflect reality in this data imbalance case. For this reason, we will also use the AuRoc metric (Definition 3) to measure if a model is able to distinguish between the two classes, in particular in the case of the imbalanced set study [16].

$$AC = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

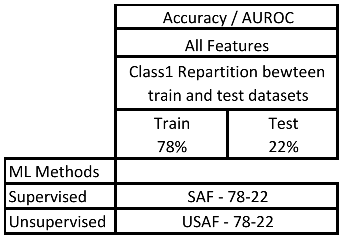
$$AuRoc = \int_0^1 \frac{TP}{P} d(\frac{FP}{N}) \tag{3}$$

where TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative.

B. ML-Methods results considering all features

In our general case study, the data were divided into 70% training data and 30% test data. So, as shown in table II, the training data contains 78% of fraudulent transactions, and 22% for the test data. We begin our experiments by studying the behavior of our two chosen metrics in the general case. This case also concerns all data features (Fig. 3).

In order to better situate the results, we start by reporting on accuracy of supervised methods. In Fig. 4, we see that accuracy of supervised algorithms is more than 99%. When



	Accuracy / AUROC	
	All Features	
	Class1 Repartition bewteen train and test datasets	
	Train 78%	Test 22%
ML Methods		
Supervised	SAF - 78-22	
Unsupervised	USAF - 78-22	

Fig. 3: General case study: all metrics and all features

dealing with such a severe imbalance data, we need to be careful when measuring model performance. Because there are only a handful of fraudulent instances, a model that predicts that each example belongs to the negative class will already achieve more than 99% accuracy. Therefore, this doesn't help us to find fraudulent cases.

Regarding the AuRoC values of supervised models, Fig. 4, we can see that the two algorithms Support Vector Machines (SVM) and Naïve Bayes (NB) have the lowest performance, which means that these models have no class separation capacity. Logistic Regression (LR), Decision Tree (DT) and Gradient Boosted Tree (GBT) models obtain a moderately good result but not the best, followed by Random Forest (RF). Finally, we can point out that Multilayer Perceptron (MLP) model obtains the best result and give the best class separation capacity.

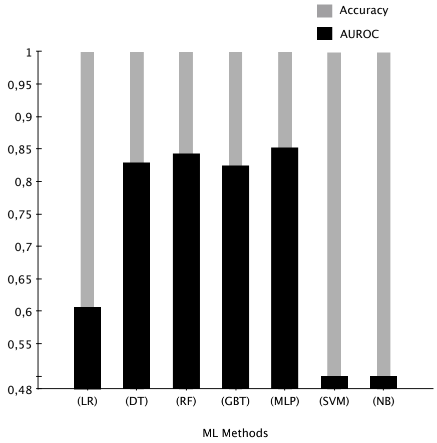


Fig. 4: SAF-78-22

Fig. 5 shows the results obtained by unsupervised models. From the point of view of accuracy measure, Bisecting K-Means obtains the worst performance, while Kmeans obtains the best accuracy. Generally, the AuRoc score is low for unsupervised algorithms.

Finally, we can say that in this general case, the supervised models give more interesting results than the unsupervised models.

C. Relevant features selection

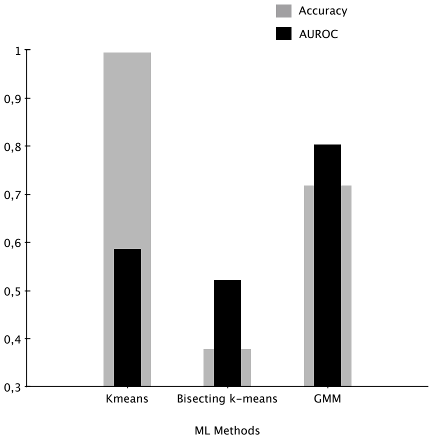


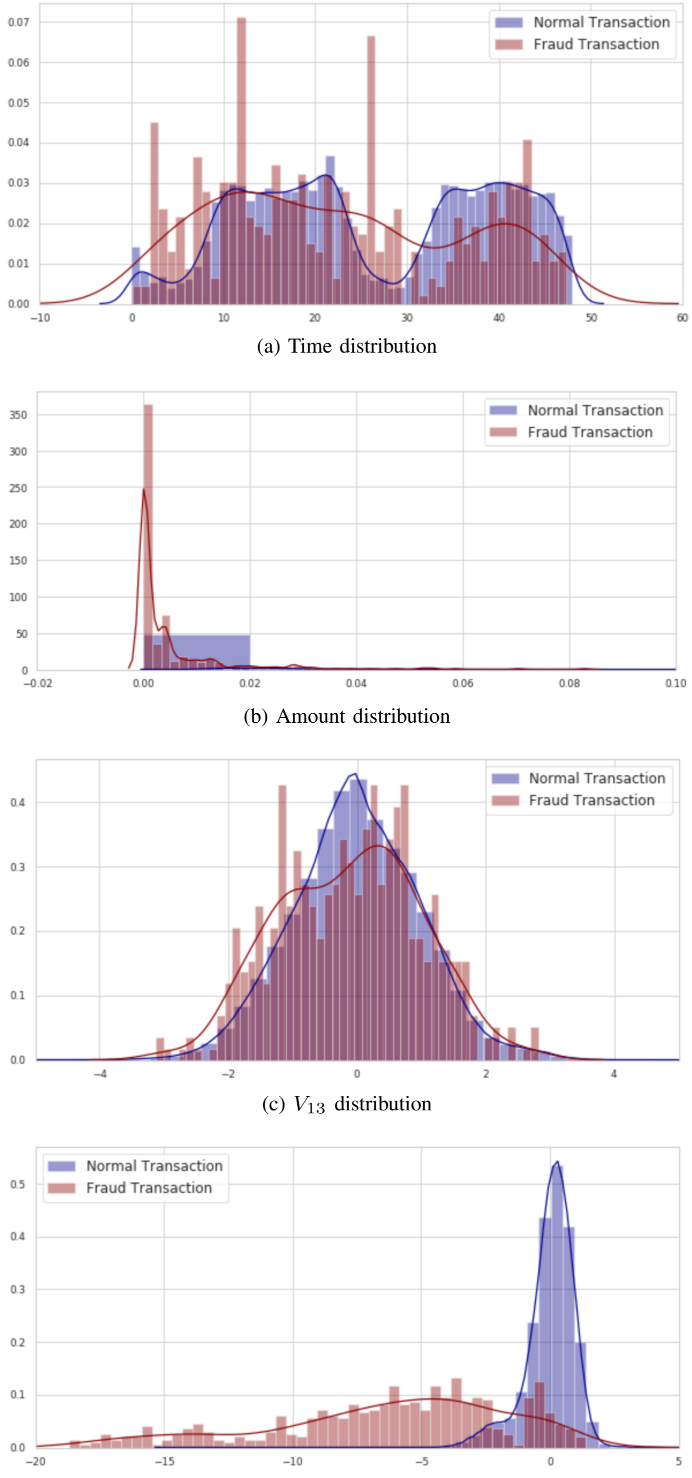
Fig. 5: USAF-78-22

In the rest of our experiments, we study the impact of the distribution of minority class data on the performance of learning algorithms. But for that, we will be interested only in the data features which are able to make a separation between the two classes (positive, negative). Formally, we select a subset of features or attributes from the set of features and eliminate redundant features that do not contribute to performance. Thus, a functionality is important when its data distributions of two classes are divergent (Definition 1). Therefore, this functionality can potentially separate the two classes and therefore improve prediction performance. Let us take the example illustrated in Fig. 6a showing the data distributions of the two classes for Time feature. We can see that the distribution of normal transactions (positive class) maps to the distribution of fraud transactions (negative class). This means that the Time feature cannot effectively contribute to the separation between the two classes. Similarly, for the Amount feature interpreted in Fig. 6b, we can observe that data distributions of the two classes are convergent. Same conclusions for the feature V^i_{13}. Consequently, these three features are irrelevant and they well be ignored by the models building. Fig. 6d shows data distributions of the two classes for V^i_{12} feature. We can see a significant divergence of two distributions, it's a feature with strong predictive power, so we can keep it during the models construction. Finally, the important features which will be considered in the following experiments are: $V^i_3, V^i_4, V^i_9, V^i_{10}, V^i_{11}, V^i_{12}, V^i_{14}, V^i_{16}, V^i_{17}, V^i_{18}, V^i_{19}$.

D. Imbalance rate study

Different experiments were conducted to study the effect of imbalance rates in the performance of the supervised and unsupervised models. Figures 7 and 10 show the different parameters considered for these experiments. Firstly, we study the supervised and unsupervised methods separately. The experiments take into account only the relevant features. For the positive class (fraudulent transactions), we studied 4 different divisions for the train and test datasets.

Predictions obtained from these experiments are evaluated in terms of Accuracy and AuRoc metrics, and presented in Figures 8, 9, 11, and 12. From Figures 8 and 9, we can observe



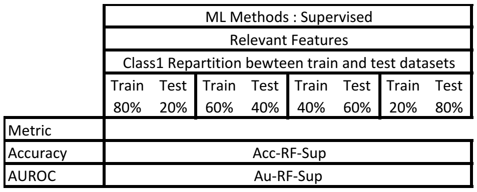
(a) Time distribution

(b) Amount distribution

(c) V_{13} distribution

(d) V_{12} distribution

Fig. 6: Selection relevant features



	ML Methods : Supervised							
	Relevant Features							
	Class1 Repartition bewteen train and test datasets							
	Train	Test	Train	Test	Train	Test	Train	Test
	80%	20%	60%	40%	40%	60%	20%	80%
Metric								
Accuracy	Acc-RF-Sup							
AUROC	Au-RF-Sup							

Fig. 7: Eval-General-4

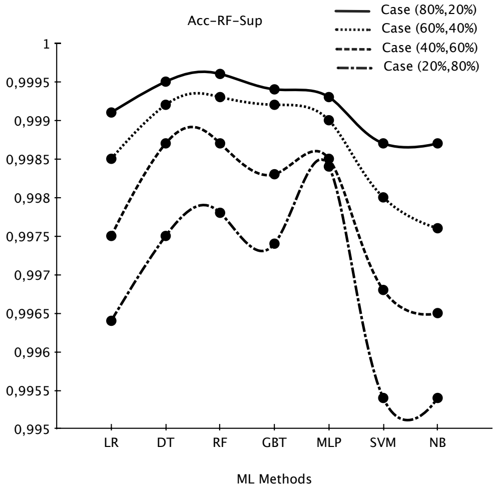


Fig. 8: Acc-RF-Sup

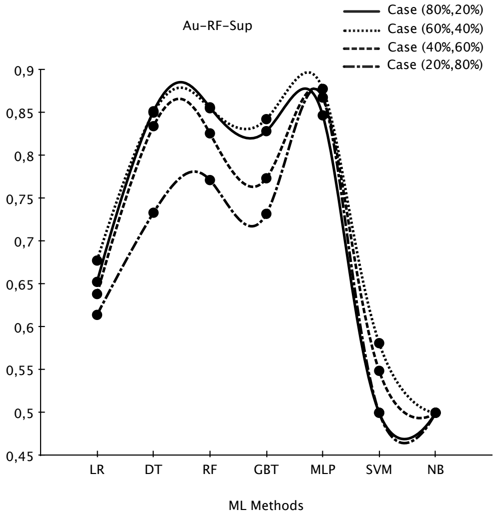
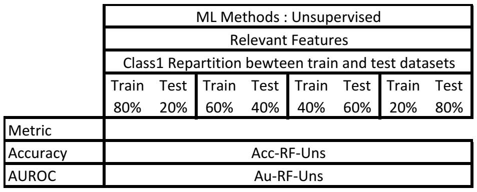


Fig. 9: Au-RF-Sup

that as imbalance rate increases in training set, Accuracy and AuRoc decrease considerably, and the usual distribution of our data, case (80%, 20%) with the relevant features, is the best to use for training the supervised models. Figures 11 and 12 show experiments results obtained by the unsupervised models. We can observe that the distribution of our data case (60%, 40%) with the relevant features is the best to use for training the unsupervised models.

VI. Conclusion and future work

In this article, we present a comparative experimental study between different techniques based on machine learning approaches for anomaly detection. We show the impact of data



	ML Methods : Unsupervised							
	Relevant Features							
	Class1 Repartition bewteen train and test datasets							
	Train	Test	Train	Test	Train	Test	Train	Test
	80%	20%	60%	40%	40%	60%	20%	80%
Metric								
Accuracy	Acc-RF-Uns							
AUROC	Au-RF-Uns							

Fig. 10: Eval-General-5

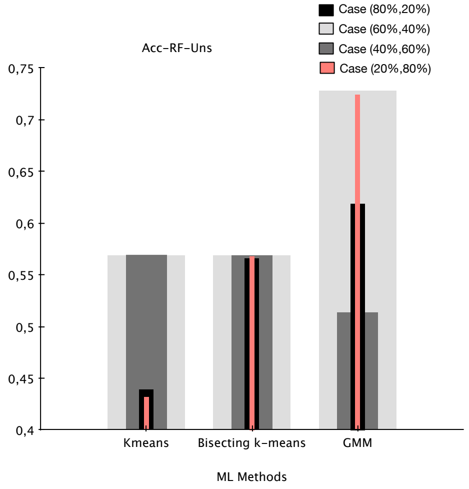


Fig. 11: Acc-RF-Uns

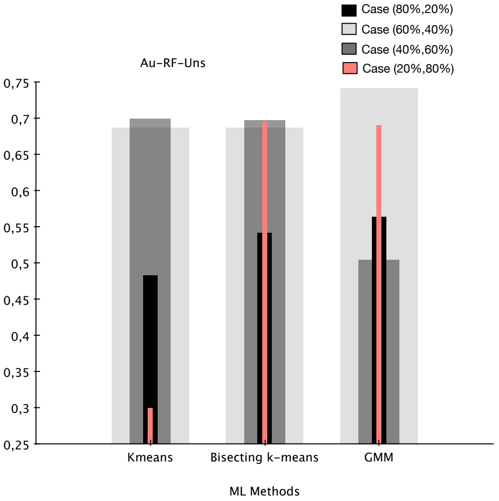


Fig. 12: Au-RF-Uns

imbalance on the performance of these approaches. Thus, from the obtained results, we note the existence of a set of parameters suitable for supervised and others for unsupervised models. Thus, our work highlights the conditions for implementing an approach for detecting anomalies in the case of a highly imbalanced dataset. However, several questions must be studied, in particular the generalization of these conditions according to data volume, their nature, etc. In our future work, we plan to address these problems and consider other approaches arising from recent developments in learning algorithms. A comparative study between our experiences and other work is also underway.

Acknowledgment

This work is carried out thanks to the support of the European Union through the PLAIBDE project of the FEDER-FSE operational program for the Nouvelle-Aquitaine region. The project is supported by aYaline company, with partners: LIAS-ENSMA laboratory in Poitiers, and the L3i laboratory at La Rochelle University.

References

[1] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pages 1–9, 2017.

[2] Luis Basora, Xavier Olive, and Thomas Dubot. Recent advances in anomaly detection methods applied to aviation. *Aerospace*, 6(11):117, 2019.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[4] P. Chuang and D. Wu. Applying deep learning to balancing network intrusion detection datasets. In *11th International Conference on Advanced Infocomm Technology (ICAIT)*, pages 213–217, 2019.

[5] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

[6] Sahil Dhankhad, Emad Mohammed, and Behrouz Far. Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In *International Conference on Information Reuse and Integration (IRI)*, pages 122–125. IEEE, 2018.

[7] Hossein Estiri and Shawn N. Murphy. A clustering approach for detecting implausible observation values in electronic health records data. *bioRxiv*, 2019.

[8] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[9] Kaggle Inc. Credit card fraud detection: Anonymized credit card transactions labeled as fraudulent or genuine, 2013.

[10] Rafiq Ahmed Mohammed, Kok-Wai Wong, Mohd Fairuz Shiratuddin, and Xuequn Wang. Scalable machine learning techniques for highly imbalanced credit card fraud detection: A comparative study. In Xin Geng and Byeong-Ho Kang, editors, *PRICAI 2018: Trends in Artificial Intelligence*. Springer, 2018.

[11] Salima Omar, Asri Ngadi, and Hamid H Jebur. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, 79(2), 2013.

[12] Shantanu Rajora, Dong-Lin Li, Chandan Jha, Neha Bharill, Om Prakash Patel, Sudhanshu Joshi, Deepak Puthal, and Mukesh Prasad. A comparative study of machine learning techniques for credit card fraud detection based on time variance. In *Symposium Series on Computational Intelligence (SSCI)*, pages 1958–1963. IEEE, 2018.

[13] Shivani Tyagi and Sangeeta Mittal. Sampling approaches for imbalanced data classification problem in machine learning. In Pradeep Kumar Singh, Arpan Kumar Kar, Yashwant Singh, Maheshkumar H. Kolekar, and Sudeep Tanwar, editors, *Proceedings of ICRIC 2019*, pages 209–221. Springer, 2020.

[14] Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. Credit card fraud detection-machine learning methods. In *International Symposium Infoteh-Jahorina*, pages 1–5. IEEE, 2019.

[15] Chunhua Wang and Dong Han. Credit card fraud forecasting model based on clustering analysis and integrated support vector machine. *Cluster Computing*, 22(6), 2019.

[16] Alice Zheng, Nicole Shelby, and Ellie Volckhausen. *Evaluating Machine Learning Models*. O'Reilly Media, Inc., 2015.