DATA SCIENCE JOURNAL

Palacio, SM. 2019. Abnormal Pattern Prediction: Detecting Fraudulent Insurance Property Claims with Semi-Supervised Machine-Learning. *Data Science Journal,* 18: 35, pp. 1–15. DOI: https://doi.org/10.5334/dsj-2019-035

RESEARCH PAPER

Abnormal Pattern Prediction: Detecting Fraudulent Insurance Property Claims with Semi-Supervised Machine-Learning

Sebastián M. Palacio

Department of Econometrics, Statistics and Applied Economics, University of Barcelona, Barcelona, ES

spalacpa9@alumnes.ub.edu

Abnormal pattern prediction has received a great deal of attention from both academia and industry, with various applications (e.g., fraud, terrorism, intrusion detection, etc.). In practice, many abnormal pattern prediction problems are characterized by the simultaneous presence of skewed data, a large number of unlabeled data and a dynamic and changing pattern. In this paper, we propose a methodology based on semi-supervised techniques and we introduce a new metric – the Cluster-Score – for fraud detection which can deal with these practical challenges. Specifically, the methodology involves transmuting unsupervised models into supervised models using the Cluster-Score metric, which defines an objective boundary between clusters and evaluates the homogeneity of the abnormalities in the cluster construction. The objectives are to increase the number of fraudulent claims detected and to reduce the proportion of claims investigated that are, in fact, non-fraudulent. The results from applying our methodology considerably improved these objectives. The experiments were performed on a real world data-set and are the results of building a fraud detection system.

Keywords: Outlier Detection; Semi-Supervised Models; Fraud; Cluster; Insurance

1. Introduction

Predicting abnormalities in environments with highly unbalanced samples and a huge mass of unlabeled data is receiving more attention as new technologies are developed (e.g., time-series monitoring, medical conditions, intrusion detection, detecting patters in images, etc.). A typical example of such a situation is provided by fraud detection (Hodge, 2004, Weiss, 2004, Phua et al., 2010, Ahuja and Singh, 2017). In general, we only have partial information about fraud cases, as well as possibly some information about false positives, that is, cases that are considered suspicious but which prove to be cases of non-fraud. The problem here is that we cannot label these cases as "non-fraud" simply because they were initially considered suspicious. For this reason, we know nothing about non-fraud cases. Moreover, fraud tends to be an outlier problem, given that we are dealing with atypical values with respect to regular data. Hence, it is likely that we only dispose of information about an extremely small sample. Yet, it so transpires, that this information is extremely useful and should not be discarded. In contrast we have a considerable amount of data that may contain fraud and or non-fraud cases and, as such, we cannot treat these data using traditional supervised algorithms.

To represent this typical case we apply an innovative semi-supervised methodology to a real fraud case. Specifically, we draw on information provided by a leading insurance company as we seek to predict fraudulent insurance claims.[1] In general terms, fraud insurance claims fall into two categories: one, those that provide only partial or untruthful information in the policy contract; and, two, those that are based on misleading or untruthful circumstances (including exaggerations). It has been estimated that cases of detected and undetected fraud represent up to 10% of all claims in Europe (The Impact of Insurance Fraud, 2013), accounting for around 10–19% of the payout bill.

[1] The study is part of the development of a fraud detection system that was implemented in 2018.

In the sector, the main services contracted are automobile and property insurance, representing 76% of total claim costs. However, while many studies have examined automobile fraud detection (see, for example, Artís et al., 1999 and 2002; Belhadji et al., 2000; Stefano and Gisella, 2001; Brockett et al., 2002; Phua et al., 2004; Viaene et al., 2007; Wilson, 2009; Nian et al., 2016), property fraud has been largely neglected, perhaps because detection is more difficult as witnesses are infrequent or they are typically cohabitants. One representative case is Bentley (2000) who uses fuzzy logic rules to detect suspicious property insurance claims in an unbalanced dataset of 98 fraudulent claims and 20,000 unknown cases. They got accuracy rates of 60% based on three artificial assumptions of 0%–5%–10% proportions of suspicious cases in the unknown claims.

In addition, private companies rarely share real fraud datasets and keep this information private to not reveal competitive details. Very small number of studies have therefore been implemented as fraud systems in insurance companies (few examples are Major and Riedinger, 1992; Cox, 1995).

Our main objective is therefore to present a variety of semi-supervised machine learning models applied to a fraud insurance detection problem. In so doing, we aim to develop a methodology capable of improving results in classification anomaly problems of this type. The key being to avoid making assumptions about the unknown fraud cases when resolving reoccurring practical problems (skewed data, unlabeled data, dynamic and changing patterns) since this can bias results.

Our reasoning for using semi-supervised models is best explained as follows. First, as pointed out by Phua, et al. (2010), **skewed data** is a challenge in many fraud studies. They find that more than 80% of the papers analyzed have a percentage of fraud cases below 30%. For instance Bentley's (2000) study have only 0.5% fraud cases whilst 99.5% are unknown, and Foster and Stine (2004) use just 2,244 cases of bankruptcies compared to 2.9 million credit card transactions to predict personal bankruptcy. Statistically speaking, fraud can be considered a case of outliers, that is, points in the data-set that differ significantly from the remaining data. Outliers do not mean noise. We refer to outliers as observations that remarkably deviate from normal data. Fraud is typically classified as abnormal behavior or a sudden change of patterns and therefore differs from noise (Barnett and Lewis, 1994; Hodge and Austin, 2004; Weiss, 2004; Aggarwal, 2015). Thus, skewed and unlabeled data is a natural consequence. Such anomalies often result from unusual events that generate anomalous patterns of activity. Were we to use unsupervised models – that is, were we to assume that we are unable to distinguish between fraudulent and non-fraudulent cases – what we defined as outliers, noise or normal data would be subjective and we would have to represent that noise as a boundary between normal data and true anomalies without any information. But, as mentioned, the number of fraud cases detected is small; however, they constitute a useful source of information that cannot be discarded.

Second, supervised models are inappropriate because, in general, we face a major problem of claim misclassifications when dealing with fraud detection (Artís et al., 2002) which could generate a substantial mass of **unknown data**. Fraud detection, typically, comprises two stages: first, it has to be determined whether the claim is suspicious or not (Viaene et al., 2007); and, second, all cases considered suspicious have to be examined by fraud investigators to determine whether the claim is fraudulent or not. This means that unsuspicious cases are never examined, which is reasonable in terms of efficiency, especially if the process cannot be automatized. Insurance adjusters have little time to perform an exhaustive investigation. Yet, the process does provide us with partial information, that is, labels for what is a small sample. Clearly, using a supervised model in this instance adds bias to the confusion matrix. Essentially, we will detect severe bias in false negatives and, therefore, many cases which are in fact fraudulent will be predicted as being non-fraudulent (Phua et al., 2004). Indeed, when using supervised algorithms we assume that the system in place is capable of discerning perfectly between fraudulent and non-fraudulent claims, an outcome that in practice is infrequent and referred to in the literature as an "omission error" (Bollinger and David, 1997; Poterba and Summers, 1995).

Finally, when fraud investigators analyze claims, they base their analysis on a small suspicious subset from previous experience and tend to compare cases to what they consider to be "normal" transactions. As data volume and the velocity of operative processes increases exponentially, human analysis becomes poorly adapted to **changing patterns** (Lei and Ghorbani, 2012).

Clearly, the information provided in relation to cases considered suspicious is more likely to be specified correctly once we have passed the first stage in the fraud detection process. This information will be useful for a part of the distribution (i.e., it will reveal if a fraudulent claim has been submitted), which is why it is very important this information be taken into account. For this reason, fraud detection in insurance claims can be considered a semi-supervised problem because the ground truth labeling of the data is partially known. Not many studies have used hybrids of supervised/unsupervised models. Williams and Huang (1997) cluster data from a Medicare Insurance, treating each cluster as a class and use them to construct a decision tree that generate decision rules. As a result, they are able to identify possible groups of interest for further investigation. Williams (1999) continues down the same line, using a system that is able to evolve with the progression of claims. Brockett et al. (1998) study automobile bodily injury insurance claims in over 387 cases. They ask loss-adjusters and investigators to group the cases by level of suspiciousness, and later use Self Organizing Maps to cluster the data and re-label it. However, basing the construction of clusters on subjective boundaries between fraud and non-fraud can bias the outcomes.

Other semi-supervised models use normal observable data to define abnormal behavioral patterns: Aleskerov et al. (1997) use past behavior as normal data to predict anomalies using Neural Networks. Kokkinaki (1997) detects atypical transactions based on users' profiles normal behavior. Murad and Pinkas (1999) identify fraudulent patterns in phone-calls finding "significant deviation" from the normal data (which is based on profiling). Kim et al. (2003) use normal product sales to detect anomalous sales patterns. However, these studies assume we have information about normal behavior, which is not always the case, and, it is questionable whether or not the normal observable data was correctly defined as normal in the first place.

We therefore seek to make three contributions to the literature: First, we apply semi-supervised techniques to an anomaly detection problem while trying to solve three combined problems: skewed data, unlabeled data and change in patterns, **without making any subjective assumption** that can bias the results. Second, we create a metric based on the logic behind the F-Score which permit us to evaluate the purity of abnormalities in the clusters. Finally, we build a fraud detection system which is applied to an actual property insurance claim fraud problem, using a real-world data-set provided by a leading insurance company.

2. Data

We use an insurance fraud data-set provided by a leading insurance company in Spain, initially for the period 2015–2016. After sanitization, our main sample consists of 303,166 property claims, some of which have been analyzed as possible cases of fraud by the Investigation Office (IO).[2]

Of the cases analyzed by the IO, 48% proved to be fraudulent. A total of 2,641 cases were resolved as true positives (0.8% of total claims) during the period under study. This means we do not know which class the remaining 99.2% of cases belong to. However, the fraud cases detected provide very powerful information, as they reveal the way in which fraudulent claims behave. Essentially, they serve as the pivotal cluster for separating normal from abnormal data.

A data lake was constructed during the process to generate sanitized data. A data lake is a repository of stored raw data, which includes structured and unstructured data in addition to transformed data used to perform tasks such as visualizing, analyzing, etc. From the data lake, we obtain 20 bottles containing different types of information related to claims. A bottle is a subset of transformed data which comes from an extract-transform-load (ETL) process preparing data for analysis. These bottles contain variables derived from the company's daily operations, which are transformed in several aspects. In total we have almost 1,300 variables. We briefly present them in **Table 1** to help explain which concepts were included in the model.

3. Methodology

If we have labeled data, the easiest way to proceed is to separate regular from outlier observations by employing a supervised algorithm. However, in the case of fraud, this implies that we know everything about the two classes of observation, i.e., we would know exactly who did and did not commit fraud, a situation that is extremely rare. In contrast, if we know nothing about the labeling, that is, we do not know who did and did not commit fraud, several unsupervised methods of outlier detection can be employed, e.g., isolation forest (Liu et al., 2008), one-class support vector machines (Schölkopf et al., 2001; Manevitz and Yousef, 2001) and elliptic envelopment (Rousseeuw and Driessen, 1999). However, they tend to be less precise and we have to assume some subjective boundary.

If, however, we have some label data about each class, we can implement a semi-supervised algorithm, such as label propagation (Zhu and Ghahramani, 2002) or label spreading (Zhou et al., 2004). Yet, these methods require that we have some information about every class in our problem, something that is not always possible. Indeed, disposing of label data information about each class is quite infrequent in certain practical problems. Additionally, we face the problem of unbalanced data, which means we rarely have clean and regular data representing the population. In fraud problems, as a norm, the data is highly imbalanced, which results in a high but biased success rate.

[2] The system applied before to detect fraud corresponds to a rule based methodology.

Table 1: The 20 Data Bottles and their descriptions extracted from a Data Lake created for this particular case study.

Bottles	Descriptions
ID	ID about claims, policy, person, etc.
CUSTOMER	Policyholder's attributes embodied in insurance policies: name, sex, age, address, etc.
CUSTOMER_PROPERTY	Customer related with the property data.
DATES	Dates of about claims, policy, visits, etc.
GUARANTEES	Coverage and guarantees of the subscribed policy.
ASSISTANCE	Call center claim assistance.
PROPERTY	Data related to the insured object.
PAYMENTS	Policy payments made by the insured.
POLICY	Policy contract data, including changes, duration, etc.
LOSS ADJUSTER	Information about the process of the investigation but also about the loss adjuster.
CLAIM	Brief, partial information about the claim, including date and location.
INTERMEDIARY	Information about the policies' intermediaries.
CUSTOMER_OBJECT_RESERVE	The coverage and guarantees involved in the claim.
HISTORICAL_CLAIM	Historical movements associated with the reference claim.
HISTORICAL_POLICY	Historical movements associated with the reference policy (the policy involved in the claim).
HISTORICAL_OTHER_POLICIES	Historical movements of any other policy (property or otherwise) related to the reference policy.
HISTORICAL_OTHER_CLAIM	Historical claim associated with the reference policy (excluding the claim analyzed).
HISTORICAL_OTHER_POL_CLAIM	Other claim associated with other policies not in the reference policy (but related to the customer).
BLACK_LIST	Every participant involved in a fraudulent claim (insured, loss-adjuster, intermediary, other professionals, etc.)
CROSS VARIABLES	Several variables constructed with the interaction between the bottles.

In the light of these issues, we propose a semi-supervised technique that can assess not only a skewed dataset problem but also one for which we have no information about certain classes. In this regard, fraud detection represents an outlier problem for which we can usually identify some, but not all, of the cases. We might, for example, have information about false positives, i.e., investigated cases that proved not to be fraudulent. However, simply because they have raised suspicions mean they cannot be considered representative of non-fraudulent cases. In short, what we usually have are some cases of fraud and a large volume of unknown cases (among which it is highly likely cases of fraud are lurking).

Bearing this in mind, we propose the application of unsupervised models so as to relabel the target variable. To do this, we use a new metric that measures how well we approximate the minority class. We can then transform the model to a semi-supervised algorithm. On completion of the relabeling process, our problem can be simplified to a supervised model. This allows us not only to set an objective boundary but to obtain a gain in accuracy when using partial information, as Trivedi et al. (2015) have demonstrated.

3.1. Unsupervised Model Selection

We start with a data-set of 303,166 cases. The original data was collected for business purposes, therefore a lot of time was put into sanitizing the data-set. It is important to remark that we set aside a 10% random subset for final evaluation. Hence, our data-set consists of 270,479 non-identified cases and 2,370 cases of fraud.

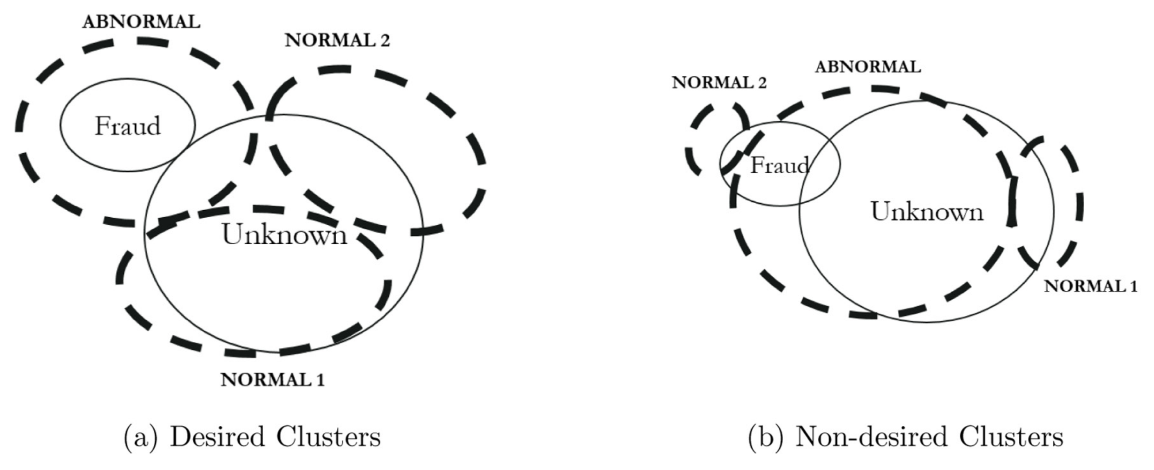
The main problem we face in this unsupervised model is having to define a subjective boundary. We have partial information about fraud cases, but have to determine an acceptable threshold at which an unknown case can be considered fraudulent. When calculating unsupervised classification models, we reduce the dimensions to clusters. Almost every algorithm will return several clusters containing mixed-type data (fraud and unknown). Intuitively, we would want the fraud points revealed to be highly concentrated into just a few clusters. Likewise, we would expect some non-revealed cases to be included with them, as in **Figure 1a**. On the other hand, we would want to avoid situations in which abnormal and normal cases are uniformly distributed between groups, as in **Figure 1b**. Thus, a limit of some kind has to be defined. But, how many of the "unknown" cases can we accept as being fraudulent?

A boundary line might easily be drawn so that we accept only cases of detected fraud or we accept every possible case as fraudulent. Yet, we know this to be unrealistic. If we seek to operate between these two extremes, intuition tells us that we need to stay closer to the lower threshold, accepting only cases of fraud and very few more, as **Figure 2** illustrates.

But once more, we do not know exactly what the correct limit is. In this way, however, we have created an experimental metric that can help us assign a score and, subsequently, define the threshold. This metric, which we shall refer to as the cluster score (CS), calculates the weighted homogeneity of clusters based on the minority and majority classes.

$$CS_{\alpha} = (1+\alpha^2)\frac{C1 * C2}{C1 + C2 * \alpha^2} \quad \text{with } \alpha > 0,\ \alpha \in \mathbf{R}$$

Essentially, it assigns a score to both the minority-class (C1) and the majority-class (C2) clusters based on the weighted conditional probability of each point. The CS expression clearly resembles the well-known F-Score,[3] which is a measure of the test's accuracy. Particularly, C1 and recall, and C2 and precision, pursue



(a) Desired Clusters

(b) Non-desired Clusters

Figure 1: Possible clusters. **(a)** shows a separable and compact cluster of the abnormal points. On the other side, **(b)** shows abnormal and normal cases uniformly distributed.

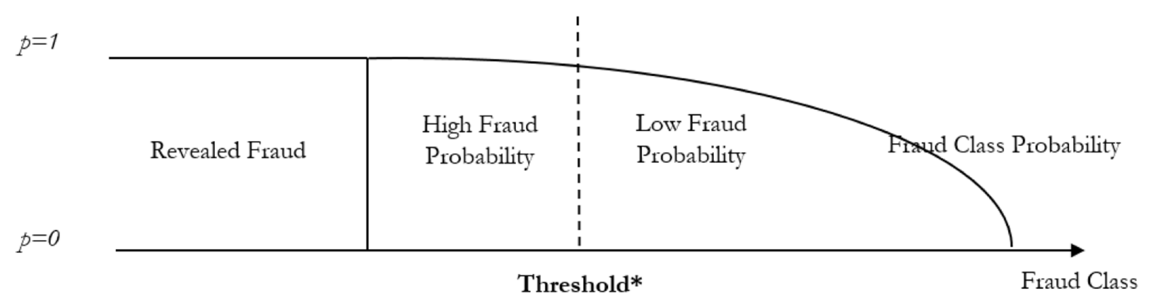


Figure 2: Schematic representation of the desired threshold which is expected to split high fraud probability cases from low fraud probability cases.

[3] F-Score is defined as $F_{\beta} = (1+\beta^2)\frac{precision * recall}{recall + \beta^2 * precision}$.

the same objectives, which in our case is to capture the maximum amount of fraud cases while also paying attention to the quality of those cases. The CS measure permits us to maximize homogeneity in the clusters. Since C1 and C2 are part of the same subset space, we have to make trade-offs (just as with recall and precision) between the optimization of C1 homogeneity and C2 homogeneity.

Moreover the α parameter allows us to maximize the homogeneity we are more concerned about. If for example, we want to obtain a more homogeneous C1 (the fraud cluster would include almost every possible case of revealed fraud), we can set a higher α, taking into account that it possibly makes the C2 homogeneity worse.

3.1.1. C1 Score

Suppose an unsupervised model generates J clusters: $\{C^1, C^2, ..., C^J\}$. The number of cases in cluster C^j is denoted by n^j.

The C1 score calculates the probability that a revealed (i.e., confirmed) fraud case belongs to cluster C^j and this probability is weighted by the total number n^j_{fraud} of fraud cases in that cluster C^j, divided by the total number of N_{fraud} of revealed fraud cases in the dataset.

$$C1 = \frac{\sum_{j=1}^{J} \frac{n^j_{fraud}}{n^j} * n^j_{fraud}}{N_{fraud}} \in [0,1]$$

Basically, we calculate the fraction of fraud cases in each cluster $j\,(n^j_{fraud}/n^j)$ and we weight these fractions by the corresponding number of fraud cases in cluster $j\,(n^j_{fraud})$.

Our objective is to maximize C1. This means ensuring all revealed fraud cases are in the same clusters. The limit C1 = 1 implies that all J clusters only contain revealed fraud cases. Therefore, we have to balance this function with another function.

3.1.2. C2 Score

C2 is the counterpart of C1. The C2 score calculates the probability that an "unknown" case belongs to cluster C^j and this probability is weighted by the total number ($n^j_{unknown}$) of unknown cases in that cluster C^j, divided by the total number of unknown cases in the data-set ($N_{unknown}$):

$$C2 = \frac{\sum_{j=1}^{J} \frac{n^j_{unknown}}{n^j} * n^j_{unknown}}{N_{unknown}} \in [0,1]$$

Notice that $n^j_{fraud} + n^j_{unknown} = n^j$. The objective is the same as that above in the case of C1: to cluster the class of unknown cases without assigning revealed fraud cases to these clusters.

3.1.3. Cluster Score

Individually maximizing C1 and C2 leaves us in an unwanted situation. Basically, they are both trying to be split. Therefore, when we maximize one, we minimize the other. If we maximize both together, this results in a trade-off between C1 and C2, a trade-off in which we can choose. Moreover, as pointed out above, we actually want to maximize C1 subject to C2. Consequently, the fraud score is constructed as follows:

$$CS_\alpha = (1+\alpha^2)\frac{C1 * C2}{C1 + C2 * \alpha^2} \quad \text{with } \alpha > 0,\ \alpha \in \mathbf{R}$$

If $\alpha = 1$, C1 and C2 will have the same weight. But if we assign $\alpha > 1$, this will reduce the weight of C2 (if $\alpha < 1$, this will reduce the weight of C1). It is important to highlight that the actual function of the cluster score is to choose between algorithms (based on the purity of the cluster construction) and α is the way to balance C1 and C2.

In conclusion, with this CS we have an objective parameter to tune the unsupervised model because it permits us to homogeneously evaluate not only different algorithms but also their parameters. While it is true that there exists a variety of internal validation indices, this metric differs in that it can also exploit information about the revealed fraud cases. That is, we take advantage of the sample that is labeled fraud to choose the best algorithm, something that internal validation indices are not able to accomplish. The only decision that remains for us is to determine the relevance of α. A numerical example can be consulted in Appendix 1.

We should stress that each time we retrieve more information about the one-class cases that have been revealed, this threshold improves. This is precisely where the entropy process of machine learning appears. As fraud is a dynamic process that changes patterns over the time, using this approach the algorithm is capable of adapting to those changes. In the one-class fraud problem discussed above, we start with an unknown distribution for which some data points are known (i.e., the fraud sample). Our algorithms, using the proposed CS metric, will gradually get closer to the best model that can fit these cases of fraud, while maintaining a margin for undiscovered cases. Now, if we obtain new information about fraud cases, our algorithms will readjust to provide the maximum CS again. As the algorithms work with notions based on density and distances, they change their shapes to regularize this new information.

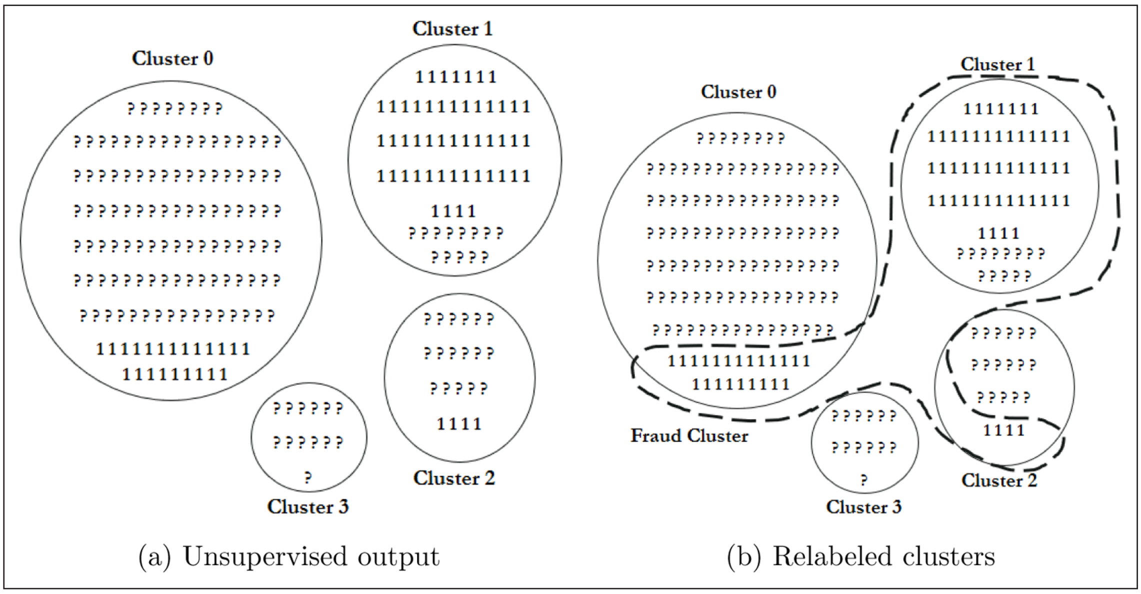
Once the best unsupervised model is attained (i.e., the model that reaches the maximum CS), we need to decide what to do with the clusters generated. Basically, we need to determine which clusters comprise fraudulent and which comprise non-fraudulent cases. The difficulty is that several clusters will be of mixed-type: e.g., minority-class points (fraud cases) and unidentified cases, as in **Figure 3a**, where the 0s are unidentified cases and the 1s are minority-class points.

In defining a threshold for a fraud case, we make our strongest assumption. Here, we assume that if a cluster is made up of more than 50% of fraud cases, this cluster is a *fraud cluster*, otherwise, it is a *non-fraud cluster*. The distinction introduced is clear: The non-fraud cluster is no longer an unidentified cluster. By introducing this assumption, we state that they are actually non-fraudulent cases. This definition acts as the key for our transition into a semi-supervised model. The assumption may seem unrealistic but, as we will see later, the best unsupervised models are capable of generating clusters with a proportion greater than 95% of fraud cases. We can, therefore be even more stringent with this assumption.

As **Figure 3b** shows, cluster 1, being composed of more than 50% fraud cases, now forms part of the more general fraud cluster, together, obviously, with the fraud cases already detected. The remaining cases that do not belong to such a dense fraud cluster are now considered non-fraud cases.

As mentioned, before applying the unsupervised algorithm, we had to make a huge effort to sanitize the original data since it was collected for business purposes. This included: handling categorized data, transforming variables, bad imputation, filtering, etc. at each bottle level. Finally, we transformed the 20 bottles at a claim level and put them together in a unique table which formed our model's input.

After that, before using this data as input, we made some important transformations. First, we filled the missing values given that many models are unable to work with them. There are simple ways to solve this,



(a) Unsupervised output (b) Relabeled clusters

Figure 3: Cluster Example Output. **(a)** shows an example of a cluster algorithm output over a sample of data points. **(b)** shows how the Cluster Score choose the points that are relabeled as fraud cases (points inside the doted line).

like using the mean or the median value of the distribution. Since we did not want to modify the original distribution, we implemented a multi-output Random Forest regressor (Breiman, 2001), to predict the missing values based on the other columns. The idea was, for each column that had missing values, we used the column as a target variable. We trained with the part without missing values, and by using the other features, we predicted the target variable.

We iterated this process in every column that had missing values (0.058% of the total values were missing). We also measured the performance of this technique using the R-squared, which is based on the residual sum of squares. Our R-squared was 89%.

Second, we normalized the data to, later, be able to apply a Principal Component Analysis (PCA), and also because many machine-learning algorithms are sensible to scale effects. Those using Euclidean distance are particularly sensitive to high variation in the magnitudes of the features. In this case, we used a robust scale approach[4] that is less affected by outliers since it uses the median value and the interpercentile ranges (we chose 90%–10%). In general, standard normalization is a widely use method. However, as in this case we are paying special attention to outliers, a mean approach might not be the best option. Outliers can often influence the sample mean/variance in a negative way. The robust scale approach removes the median and scales the data according to a quantile range. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Median and interquartile range are then stored to be used on test/new data.

Third, we applied Principal Component Analysis to resolve the high dimensionality problem (we had almost 1,300 variables). This method reduces confusion in the algorithms and solves any possible collinearity problems. PCA decomposes the data-set in a set of successive orthogonal components that explain a maximum amount of the data-set's variance. When setting a data-set's variance threshold, a trade-off between over-fitting and getting the variation in the data-set is made. We chose a threshold of 95% (recommended threshold is between 95% and 99%), which resulted in 324 components. After this transformations, the unsupervised algorithm can thus be summarized as seen in **Algorithm 1**.

The main reason is that it has a low noise sensitivity as it ignores small variations in the background (based on a maximum variation basis). While it is true that there are several non-linear formulations for dimensionality reduction that may get better results, some studies have actually found that non-linear techniques are often not capable of outperforming PCA. For instance Van Der Maaten et al. (2009) compared PCA versus twelve non-linear dimensionality reduction techniques on several data-sets and they couldn't conclude that the non-linear techniques outperformed PCA.

Algorithm 1: Unsupervised algorithm

```
Data: Load transformed data-set. Oversample the fraud cases in order to have the same amount as the number of unknown cases.
for $k \in K = \{model_1, model_2, ...\}$ where K is a set of unsupervised models. do
    for $i \in I$ where I is a matrix of parameter vectors containing all possible combinations of the parameters in model k do
        We fit the model $k$ with the parameters $i$ to the oversampled data-set.
        We get the $J$ clusters: $\{C^1, C^2, ..., C^J\}$ for the combination $\{k, i\}$, i.e., $C_{k,i} = \{C^1_{k,i}, C^2_{k,i}, ..., C^J_{k,i}\}$
        For $C_{k,i}$ we calculate C1 Score and C2 Score and we obtain the cluster score $CS_{k,i}$, based on the acceptance threshold $t^*$.
        Save the cluster score result $CS_{k,i} \in CS_{K,I}$, where $CS_{K,I}$ is the cluster score vector for each pair $\{k, i\}$.
    end
end
Choose the optimal $CS^*$ where $CS^* = max\{CS_{K,I}\}$
Relabel the fraud variable using the optimal clustering model derived from $CS^*$. Each unknown case in a fraud cluster is now equal to 1, known fraud cases are equal to 1 and remaining cases are equal to 0.
```

[4] We use the formulation $Z = (x - x_{median})/(p_{90} - p_{10})$.

3.2. Supervised Model Selection

We now have a redefined target variable that we can continue working with by applying an easy-to-handle supervised model. The first step involves re-sampling the fraud class to avoid unbalanced sample problems. Omitting this step, means that our model could be affected by the distribution of classes, the reason being that classifiers are in general more prone to detect the majority class rather than the minority class. We, therefore, oversample the data-set to obtain a 50/50 balanced sample. We use two oversampling methods, Adaptive Synthetic Sampling Approach (ADASYN) by He et al. (2008), and balanced subsampling. ADASYN finds the n-nearest neighbors in the minority class for each of the samples in the class. It creates random samples from the connections and adds a random small value to the points in order to scatter them and break the linear correlation with the parent point. The balanced subsample method on the other hand, does not need to create synthetic points since the samples used are already balanced. The balanced samples are obtained by using weights inversely proportional to class frequencies for each iteration in a supervised tree based algorithm.

The second step, involves conducting a grid search and a Stratified 5-fold cross-validation (CV) based on the F-Score[5] to obtain the optimal parameters for three different models: extreme randomized tree -ERT- (Geurts et al., 2006), gradient boosting -GB- (Freund and Schapire, 1996) and a light XGB -LXGB- (Ke et al., 2017). Cross-validation is a great way to avoid over-fitting, i.e., failing to predict new data. We train using $k-1$ folds (data subsets) and we validate our model by testing it on the remaining fold. To prevent an imbalance problem in the folds, we use a stratified k-folds strategy which returns subsets containing approximately the same distribution of classes as the original data-set.

We have to be careful not to over-fit the model during the cross-validation process, particularly when using oversampling methods. Step one and step two, therefore have to be executed simultaneously. Oversampling before cross-validating would generate samples that are based on the total data-set. Consequently, for each $k-1$ training fold, we would include very similar instances in the remaining test fold, and vice versa. This is resolved by first, stratifying the data, and then oversampling the $k-1$ folds, without taking into account the validation fold. Finally, we concatenate all the predictions.

Additionally, we combine the supervised models using stacking models. Stacking models is combining different classifiers, applied to the same data-set, and getting different predictions that can be “stacked” up to produce one final prediction model. The idea is very similar to k-fold cross validation, dividing the training set into several subsets or folds. For all $k-1$ folds, predictions are obtained by using all the supervised models (called the base models). The predictions are stored to be used as features for the stacking model in the full training data-set. Finally, a new model (the stacking model or the Meta model) is fitted to the improved data-set. The stacking model can discern whether a model performs well or poorly, which is very useful since one model might have high performance when predicting fraud, but not when predicting non-fraud, and vice versa. The combination of both could therefore improve the results. We try three different ways of combining classifiers, modifying the Meta model: GB and LXGB with Meta ERT, GB and ERT with Meta LXGB, and LXGB and ERT with Meta GB.

Once we have the optimal parameters for each model, we calculate the optimal threshold that defines the probability of a case being fraudulent or non-fraudulent, respectively.

Finally, we identify the two models that perform best on the data-set – the best acting as our main model implementation, the other controlling that the predicted claims are generally consistent. The algorithm can be summarized as seen in **Algorithm 2**.

4. Results

4.1. Performance

Table 2 shows the main unsupervised modeling results of the tuning process. We tried different combinations of distance based models, density based models and outlier models: Mini-Batch K-Means (Sculley, 2010), Isolation Forest (Liu, 2008), DBSCAN (Ester et al., 1996), Gaussian Mixture and Bayesian Mixture (Figueiredo and Jain, 2002). Mini-batch K-Means is not only much faster than the other models, it also provides the best results. It is similar to K-Means++, both using the Euclidean distance between points as the objective function, however it can also reduce computation time. Subsets of the input data are taken and randomly sampled in each iteration, converging more quickly to a local solution.

[5] The F-Score was constructed using $\beta = 2$, as we needed to place greater weight on the recall.

Algorithm 2: Supervised algorithm.

```
Data: Load relabeled data-set.
for $model_i \in M' = \{M, S\}$ where M is the set of supervised individual models M and S the set of stacking models from M do
    for $\{train_k, test_k\}$ folds in the Stratified k-Folds do
        We apply PCA to folder traink and save the weights/parameters.
        if Oversample==True then $train'_k = oversample(train_k)$ where oversampling is applied to 50/50 using the ADASYN method.
        else $train'_k = train'_k$ and the balanced subsampling option is activated.
        Fit the $model_i$ in $train'_k$, where $model_i \in M' = \{M, S\}$.
        Transform $test_k$ with PCA's weights/parameters and get predicted probabilities $p_k$ of $test_k$ using $model_i$.
        Save the probabilities $p_k$ in $P_i$, where $P_i$ is the concatenation of $model_i$'s probabilities.
    end
    for $\forall t_i \in [0, 1]$, where t is a probability threshold of the $model_i$ to consider a case as fraudulent do
        if $P_i \geq t_i$ then $P_i = 1$
        else $P_i = 0$
        Using $P_i$, where now $P_i$ is a binary list, we calculate,
            $FScore_{i,t} = (1+\beta^2) * \frac{precision * recall}{recall + \beta^2 * precision}$ with $\beta = 2$.
        Save $FScore_{i,t}$ in $FScore_i$, a list of vectors of $model_i$ with FScore results for each t.
    end
    We get $FScore_i^* = max\{FScore_i(t)\}$.
end
```

Table 2: Unsupervised model results.

Model	n Clusters	C1	C2	CS ($\alpha = 2$)
Mini-Batch K-Means	4	96.6%	96.6%	96.6%
Isolation Forest	2	51.5%	51.1%	51.4%
DBSCAN	2	50.2%	49.8%	50.1%
Gaussian Mixture	5	95.0%	95.0%	96.3%
Bayesian Mixture	6	96.5%	96.4%	96.5%

C1 indicates that the minority-class (fraud) clusters comprise approximately 96.59% of minority data points on a weighted average. In contrast, C2 indicates they are made up of 96.59% of unknown cases. As can be seen in **Table 3**, more than 95% of the cases in the central cluster are fraudulent (well above our 50% assumed threshold), but it also contains an additional 6,047 unknown cases (Cluster 0 now contains an additional 5,890 cases, and Cluster 1 an additional 157 cases). This is our core fraud cluster and the one we use when renaming the original labels.

After relabeling the target variable (with the Mini-Batch K-Means output), we calculate the supervised models performance using Stratified 5-Fold CV on the data-set. The results of each of the supervised models and of the stacking models is shown in **Table 4**.

As can be appreciated, we have two recall values. The cluster recall is the metric derived when using the relabeling target variable. The original recall emerges when we recover the prior labeling (1 if it was fraud,

Table 3: Oversampled Unsupervised Mini-Batch K-Means.

Clusters	Fraud	Percentage
0	0	2%
0	1	98%
1	0	99%
1	1	1%
2	0	100%
2	1	0%
3	0	1%
3	1	99%

Table 4: Supervised model results.

Model	Cluster Recall	Original Recall	Precision	F-Score
ERT-ss	0.9734	0.9840	0.6718	0.8932
ERT-os	0.9647	0.9819	0.6937	0.8948
GB	0.9092	0.9376	0.6350	0.8369
LXGB	0.8901	0.9249	0.7484	0.8576
Stacked-ERT	0.8901	0.9283	0.7524	0.8587
Stacked-GB	0.8947	0.9287	0.7630	0.8649
Stacked-LXGB	0.9180	0.9464	0.6825	0.8588

0 otherwise). As can be seen, the results are strikingly consistent. We are able to predict fraud cluster with a recall of up to 89–97% in every case. But, more impressively yet, we can capture the original fraud cases with a recall close to 98%. The precision is slightly lower, but in almost all cases it is higher than 67%. These are particularly good results for a problem that began as an unsupervised high-dimensional problem with an extremely unbalanced data-set.

The two best models are both extreme randomized trees: the first uses balanced subsampling -ERT-ss- (i.e., for every random sample used during the iteration of the trees, the sample is balanced by using weights inversely proportional to class frequencies), and serves here as our base model; the second uses an ADASYN oversampling method-ERT-os- and serves as our control model.

4.2. Investigation Office Validation

At the outset, we randomly set aside 10% of the data (30,317 claims). In this final step, we want to go further and examine these initial claims as test data. Our results are shown in **Table 5**.

As can be appreciated, the control model (**Table 5b**) has a recall of 97% while the base model (**Table 5a**) has an impressive recall of 100%. However, the real added value depends on the non-investigated fraud cases, i.e., cases not previously detected but which would boost our results if shown to be fraudulent (non-investigated predicted as fraud). We, therefore, sent these cases to the IO for analysis.

The IO investigated 367 cases (at the intersection between the ERT-ss and ERT-os models). Two fraud investigators analyzed each of these cases, none of which they had previously seen as the rule model had not detected them.

Of these 367 cases, 333 were found to present a very high probability of being fraudulent. This means that only 34 could be ruled out as not being fraudulent. Recall that from the original sample of 415 cases, the fact that 333 presented indications of fraud means we have a precision of 88%. In short, we managed to increase the efficiency of fraud detection by 122.8%. These final outcomes are summarized in **Table 6**.

4.3. Dynamic Learning

One of the challenges in fraud detection is that it is a dynamic process which can change its patterns over time. A year later, we retest the model with new data. We now have 519,921 claims to evaluate. We initially start out with a similar proportion of fraud cases (0.88%)- we are now able to train with 4,623 fraud cases to further improve results.

First, we recalculate the unsupervised algorithm, getting a Cluster-Score of 96.89%. As can be seen in **Table 7**, Cluster 0 contains almost every normal case. On the other hand, we can clearly distinguish two fraud clusters: Cluster 1, in which 99.31% are fraud cases, and Cluster 2, in which 97.36% are fraud cases. Our 50% threshold therefore becomes insignificant again.

Using the Extreme Randomized Subsampled approach (ERT-ss) and the Extreme Randomized oversampled with ADASYN (ERT-os), and the Stratified 5-fold cross validation approach we retrain the model. **Table 8** shows the main results.

The base model greatly improves the homogeneity of the fraud and non-fraud clusters. In particular, it provides a gain of 33% in the precision score and of 6.2–6.8% in the F-Score.

Table 5: Model Robustness Check.

Original Value	Prediction	Cases
Non-Investigated	Non-Fraud	29.631
Fraud	Non-Fraud	0
Non-Investigated	Fraud	415
Fraud	Fraud	271

(a) ERT-ss Robustness Check

Original Value	Prediction	Cases
Non-Investigated	Non-Fraud	29.656
Fraud	Non-Fraud	8
Non-Investigated	Fraud	390
Fraud	Fraud	263

(b) ERT-os Robustness Check

Table 6: Base Model Final Results.

Original Value	Prediction	Cases
Non-Investigated	Non-Fraud	29.631
Fraud	Non-Fraud	0
Non-Fraud	Fraud	(415 – 333) = 82
Fraud	Fraud	(271 + 333) = 604

Table 7: Oversampled Unsupervised Mini-Batch K-Means.

Clusters	Fraud	Percentage
0	0	99.4%
0	1	0.6%
1	0	0.7%
1	1	99.3%
2	0	2.6%
2	1	97.4%

Table 8: Base Model with the machine-learning process applied.

Period	Jan 15–Jan 17	Jan 15–Jan 18
Claims	303,166	519,921
Observed Fraud	2,641	4,623
Cluster Score	96.59%	96.89%
Recall Score ERT-ss	97.34%	96.31%
Precision Score ERT-ss	67.18%	89.35%
F-Score ERT-ss	89.32%	94.84%
Recall Score ERT-os	96.47%	96.44%
Precision Score ERT-os	69.37%	92.18%
F-Score ERT-os	89.48%	95.56%

5. Conclusion

This paper has sought to offer a solution to the problems that arise when working with highly unbalanced data-sets for which the labeling of the majority of cases is unknown. In such cases, we may dispose of a few small samples that contain highly valuable information. Here, we have presented a fraud detection case, drawing on the data provided by a leading insurance company, and have tested a new methodology based on semi-supervised fundamentals to predict fraudulent property claims.

At the outset, the Investigation Office (IO) did not investigate many cases (around 7,000 cases from a total of 303,166). Of these, only 2,641 were actually true positives (0.8% of total claims), with a success rate of 48%. Thanks to the methodology devised herein, which continuously readapts to dynamic and changing patterns, we can now investigate the whole spectrum of cases automatically, obtaining a total recall of 96% and a precision of 89–92%. In spite of the complexity of the initial problem, where the challenge was to detect fraud dynamically without knowing anything about 99.2% of the sample, the methodology described has been shown to be capable of solving the problem with great success.

Additional File

The additional file for this article can be found as follows:

- **Appendix.** Practical Example. DOI: https://doi.org/10.5334/dsj-2019-035.s1

Acknowledgement

The author would like to thank to Cristina Rata and Joan-Ramon Borrell for constructive criticism of the manuscript.

Competing Interests

The author has no competing interests to declare.

References

Aggarwal, CC. 2015. Outlier analysis. In *Data mining*. Springer, pp. 237–263. DOI: https://doi.org/10.1007/978-3-319-14142-8_8

Ahuja, MS and **Singh, L.** 2017. Online fraud detection-a review. *International Research Journal of Engineering and Technology*, 4(7): 2509–2515.

Aleskerov, E, Freisleben, B and **Rao, B.** 1997. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Computational Intelligence for Financial Engineering (CIFEr)*. IEEE, pp. 220–226.

Barnett, V and **Lewis, T.** 1994. *Outliers in statistical data*. Wiley Chichester.

Belhadji, EB, Dionne, G and **Tarkhani, F.** 2000. A model for the detection of insurance fraud. *The Geneva Papers on Risk and Insurance-Issues and Practice*, 25(4): 517–538. DOI: https://doi.org/10.1111/1468-0440.00080

Bentley, PJ. 2000. Evolutionary, my dear Watson investigating committee-based evolution of fuzzy rules for the detection of suspicious insurance claims. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., pp. 702–709.

Bollinger, CR and **David, MH.** 1997. Modeling discrete choice with response error: Food stamp participation. *Journal of the American Statistical Association*, 92(439): 827–835. DOI: https://doi.org/10.1080/01621459.1997.10474038

Breiman, L. 2001. Random forests. *Machine learning*, 45(1): 5–32. DOI: https://doi.org/10.1023/A:1010933404324

Brockett, PL, Derrig, RA, Golden, LL, Levine, A and **Alpert, M.** 2002. Fraud classification using principal component analysis of ridits. *Journal of Risk and insurance*, 69(3): 341–371. DOI: https://doi.org/10.1111/1539-6975.00027

Brockett, PL, Xia, X and **Derrig, RA.** 1998. Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *Journal of Risk and Insurance*, pp. 245–274. DOI: https://doi.org/10.2307/253535

Cox, E. 1995. A fuzzy system for detecting anomalous behaviors in healthcare provider claims. *Intelligent Systems for Finance and Business*, pp. 111–134.

Ester, M, Kriegel, H-P, Sander, J and **Xu, X.** 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, Vol. 96, pp. 226–231.

Europe, I. 2013. The impact of insurance fraud. Brussels: Insurance Europe.

Figueiredo, MAT and **Jain, AK.** 2002. Unsupervised learning of finite mixture models. *Transactions on pattern analysis and machine intelligence*, 24(3): 381–396. DOI: https://doi.org/10.1109/34.990138

Foster, DP and **Stine, RA.** 2004. Variable selection in data mining: Building a predictive model for bankruptcy. *Journal of the American Statistical Association*, 99(466): 303–313. DOI: https://doi.org/10.1198/016214504000000287

Freund, Y and **Schapire, RE.** 1996. Experiments with a new boosting algorithm. In *ICML*, Vol. 96. Citeseer, pp. 148–156.

Geurts, P, Ernst, D and **Wehenkel, L.** 2006. Extremely randomized trees. *Machine learning*, 63(1): 3–42. DOI: https://doi.org/10.1007/s10994-006-6226-1

He, H, Bai, Y, Garcia, EA and **Li, S.** 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IJCNN (World Congress on Computational Intelligence)*. IEEE, pp. 1322–1328.

Hodge, V and **Austin, J.** 2004. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2): 85–126. DOI: https://doi.org/10.1023/B:AIRE.0000045502.10941.a9

Ke, G, Meng, Q, Finley, T, Wang, T, Chen, W, Ma, W, Ye, Q and **Liu, T-Y.** 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pp. 3146–3154.

Kim, J, Ong, A and **Overill, RE.** 2003. Design of an artificial immune system as a novel anomaly detector for combating financial fraud in the retail sector. In *Evolutionary Computation*, Vol. 1. IEEE, pp. 405–412.

Kokkinaki, AI. 1997. On atypical database transactions: identification of probable frauds using machine learning for user profiling. In *Knowledge and Data Engineering Exchange Workshop, 1997. Proceedings*. IEEE, pp. 107–113.

Lei, JZ and **Ghorbani, AA.** 2012. Improved competitive learning neural networks for network intrusion and fraud detection. *Neurocomputing*, 75(1): 135–145. DOI: https://doi.org/10.1016/j.neucom.2011.02.021

Liu, FT, Ting, KM and **Zhou, ZH.** 2008. Isolation forest. In *International Conference on Data Mining*. IEEE, pp. 413–422. DOI: https://doi.org/10.1109/ICDM.2008.17

Major, JA and **Riedinger, DR.** 1992. A hybrid knowledge/statistical-based system for the detection of fraud. *International Journal of Intelligent Systems*, 7(7): 687–703. DOI: https://doi.org/10.1002/int.4550070709

Manevitz, LM and **Yousef, M.** 2001. One-class svms for document classification. *Journal of Machine Learning research*, 2: 139–154.

Murad, U and **Pinkas, G.** 1999. Unsupervised profiling for identifying superimposed fraud. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, pp. 251–261. DOI: https://doi.org/10.1007/978-3-540-48247-5_27

Nian, K, Zhang, H, Tayal, A, Coleman, T and **Li, Y.** 2016. Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science*, 2(1): 58–75. DOI: https://doi.org/10.1016/j.jfds.2016.03.001

Phua, C, Alahakoon, D and **Lee, V.** 2004. Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter*, 6(1): 50–59. DOI: https://doi.org/10.1145/1007730.1007738

Phua, C, Lee, V, Smith, K and **Gayler, R.** 2010. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

Rousseeuw, PJ and **Driessen, KV.** 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3): 212–223. DOI: https://doi.org/10.1080/00401706.1999.10485670

Schölkopf, B, Platt, JC, Shawe-Taylor, J, Smola, AJ and **Williamson, RC.** 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7). DOI: https://doi.org/10.1162/089976601750264965

Sculley, D. 2010. Web-scale k-means clustering, In *Proceedings of the 19th international conference on World wide web*. ACM, pp. 1177–1178. DOI: https://doi.org/10.1145/1772690.1772862

Stefano, B and **Gisella, F.** 2001. Insurance fraud evaluation: a fuzzy expert system. In *Fuzzy Systems, 2001. The 10th IEEE International Conference*, Vol. 3. IEEE, pp. 1491–1494.

Trivedi, S, Pardos, ZA and **Heffernan, NT.** 2015. The utility of clustering in prediction tasks. *arXiv preprint arXiv:1509.06163*.

Van Der Maaten, L, Postma, E and **Van den Herik, J.** 2009. Dimensionality reduction: a comparative. *Journal of Machine Learning Research*, 10(66–71): 13.

Viaene, S, Ayuso, M, Guillen, M, Van Gheel, D and **Dedene, G.** 2007. Strategies for detecting fraudulent claims in the automobile insurance industry. *European Journal of Operational Research*, 176(1): 565–583. DOI: https://doi.org/10.1016/j.ejor.2005.08.005

Weiss, GM. 2004. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1): 7–19. DOI: https://doi.org/10.1145/1007730.1007734

Williams, GJ. 1999. Evolutionary hot spots data mining. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 184–193. DOI: https://doi.org/10.1007/3-540-48912-6_26

Williams, GJ and **Huang, Z.** 1997. Mining the knowledge mine. In *Australian Joint Conference on Artificial Intelligence*. Springer, pp. 340–348. DOI: https://doi.org/10.1007/3-540-63797-4_87

Wilson, JH. 2009. An analytical approach to detecting insurance fraud using logistic regression. *Journal of Finance and Accountancy*, 1: 1.

Zhou, D, Bousquet, O, Navin Lal, T, Weston, J and **Scholkopf, B.** 2004. Learning with local and global consistency. In *Advances in neural information processing systems*, pp. 321–328.

Zhu, X and **Ghahramani, Z.** 2002. Learning from labeled and unlabeled data with label propagation. *Technical report, CMU-CALD-02-107*. Carnegie Mellon University.

How to cite this article: Palacio, SM. 2019. Abnormal Pattern Prediction: Detecting Fraudulent Insurance Property Claims with Semi-Supervised Machine-Learning. *Data Science Journal,* 18: 35, pp. 1–15. DOI: https://doi.org/10.5334/dsj-2019-035

Submitted: 06 December 2018 **Accepted:** 03 July 2019 **Published:** 17 July 2019

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS