An application of unsupervised fraud detection to Passenger Name Records

Rémi Domingues, Francesco Buonora, Romain Senesi, Olivier Thonnard

An application of unsupervised fraud detection to Passenger Name Records

Rémi Domingues, Francesco Buonora, Romain Senesi, Olivier Thonnard
Amadeus, France
name.surname@amadeus.com

***Abstract*—Fraud is a threat that most online service providers must address in the development of their systems to ensure an efficient security policy and the integrity of their revenue. If rule-based systems and supervised methods usually provide the best detection and prevention, labelled training datasets are often non-existent and such solutions lack reactivity when facing adaptive fraudsters. Many generic fraud detection solutions have been made available for companies though cannot compete with dedicated internal implementations.**

This study presents an evaluation of some of the most widely used machine learning algorithms for unsupervised fraud detection applied to travel booking information represented by Passenger Name Records (PNR). The current paper also highlights the use of some aggregation functions relying on fuzzy logic and interpolation as an extension of unsupervised ensemble learning.

Keywords: fraud detection, outlier detection, unsupervised learning, Passenger Name Record.

1. Introduction

The reservation system of Amadeus, a Global Distribution System (GDS) providing several platforms connecting the travel ecosystem, is targeted by fraud attempts that could lead to revenue losses and indemnifications. Those fraud attempts are supposedly performed by travel agencies in charge of flight booking operations and are motivated by a potential access to undeserved advantages, such as unmerited financial incentives. To our knowledge, this paper is the first one applying unsupervised fraud detection to Passenger Name Records (PNR) and benefits from the dataset of one of the leading GDS managing almost half of the flight bookings worldwide.

Despite the continuous progress in securing communication methods and information systems, the complexity of most systems and the ingenuity of some fraudsters allow new forms of fraud to be performed every day. This fact added to the critical value of money flows and company assets has made fraud detection an active research area making extensive use of machine learning techniques.

A possible approach to fraud detection relies on supervised learning to block fraud attempts based on fraudulent and non-fraudulent samples. A class of widely used algorithms rely on rule-based detection, automatically inferring discriminative rules from a labelled training set. Fawcett at al. [5] describe an automatic rule generation algorithm applied to cellular cloning performing better than manually generated rules based on expert knowledge. Other techniques based on decision trees [6] or Support Vector Machine (SVM) algorithms provide good results with more than 90% of correct classification rate when applied to mobile phone fraud detection, especially when combined using ensemble methods such as bagging or boosting as shown in [11]. The efficiency of artificial neural networks for fraud detection has been demonstrated by Ghosh et al. using credit card transactions as input of a feed-forward RBF network [7]. When a fraud-free dataset is available, supervised novelty detection techniques like one-class SVM can be used, fitting a novelty boundary to a given class using kernel methods and a parametric slack. This algorithm is detailed in section 3.1

Nevertheless, a labelled dataset is not available in many real-world applications which prevents the use of supervised learning. Supervised algorithms may besides suffer from unbalanced class sizes resulting in a poor detection. In addition, such techniques cannot identify new fraud patterns and will thus be ineffective at stopping uncovered fraud behaviors. Unsupervised methods can contribute to reduce the delay between a new flaw detection and its resolution and thus grant a strong competitive advantage to targeted companies.

Unsupervised learning allows the discovery of suspicious behaviors and do not require any prior knowledge on verified fraudulent cases. It is the result of a preliminary learning step modelling an expected standard behavior followed by an outlier detection step from which anomalies can be detected.

Among unsupervised techniques, probabilistic methods estimate the probability density function (PDF) of a dataset. Such algorithms include Gaussian Mixture Model (GMM) [14] which can be an input for multivariate novelty detection in extreme value theory (EVT) [3]. Kernel density estimation (KDE), also known as Parzen window, is a nonparametric technique which has been successfully applied to network intrusion detection [20]. Gaussian Processes is another probabilistic example and it allows one-class classification for novelty detection [16].

When dealing with temporal event sequences, Hidden Markov Model (HMM) has proven to be an efficient state-space model to represent a system. It enables the computation of the likelihood for a given pattern which can be thresholded to reject sequences of low probability [21].

Distance-based methods such as clustering can be used to characterise normal classes by a sufficient number of data points close to each other. Among some well-known clustering algorithms for outlier and novelty detection is the k-Nearest Neighbors (kNN) algorithm, but also k-means [10] and some of its extensions such as fuzzy c-means.

As part of non-distance based methods, Guha et al. have developed a robust hierarchical clustering algorithm called ROCK [8], which employs *links* instead of distances when merging clusters. ROCK clustering and similar non-distance techniques extend to non-metric similarity measures that are relevant in situations where some notion of domain similarity represented by links (or *relationships*) is the only source of knowledge.

Artificial neural networks provide also ways to identify outliers, either by using a Self-Organizing Map (SOM) as demonstrated in [15], or by thresholding the value of the energy function output by a Hopfield network in [4].

This paper focuses on the Passenger Name Record (PNR) standard described in section 2 and used by Global Distribution Systems (GDS) to store and exchange traveller information and requests on passenger itineraries. We describe a benchmark of several unsupervised fraud detection methods applied to multivariate data pertaining to PNR's, including but not limited to Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), density clustering (DBSCAN), hierarchical clustering and Self-Organizing Maps (SOM). Several aggregation operators related to multi-criteria decision analysis (MCDA) are also compared in the context of ensemble methods. We show that combining algorithms using MCDA operators gives greater results than any single algorithm.

2. Dataset

A Passenger Name Record (PNR) is a database record containing information related to the itinerary of one or more passengers created by travel operators (e.g. travel agents, online travel agencies), travel providers (e.g. airlines), a third-party GDS or computer reservation systems (CRS). It is mostly used for the booking and check-in procedures and can target itineraries containing several flight segments. It includes but is not limited to information about passengers (names, addresses), frequent flyer status, flight segments (IATA codes, flights schedules), special service requests (seats, luggage, meals), tickets, forms of payment, the travel agents source of the booking or modifications and references to other PNRs for split group records. This record is divided in an ordered list of envelopes following the EDIFACT standard [9], each of them describing the current state of the PNR and the list of changes applied during the last transaction. Most frauds can only be achieved by performing a specific sequence of actions and thus require to study the complete modification history to be detected.

The dataset used by our experiment is a random sample of 40,000 PNRs created in 2015 and containing a total of 850,000 envelopes weighting 20GB. 58 relevant features have been defined by PNR and fraud experts to be extracted per envelope and to represent the most important operations. Most of them are either timestamps (e.g. creation date of an envelope) or counters covering the number of points of sale, the changes applied to passengers, frequent traveller cards, segments (segment marriages, special service requests) and forms of payment (FoP). The complete list of actions performed in the envelope is also extracted though does not follow the original sequence order. To work at PNR level, we aggregate the envelopes into a single feature vector of 83 features per PNR using a Hadoop Map-Reduce job. The aggregation process computes the number of envelopes, the age of the PNR and aggregates most features using maximums, sums, averages, standard deviations and ratios (e.g. $\frac{final\ number\ of\ segments}{sum\ of\ added\ segments}$). 82 features are thus numeric while the 83^{rd} feature is the concatenated list of actions. The distribution of a subset of aggregated features is described in Table 1 with quantiles and other standard statistical measures. It shows strong outliers and already suggests some potential frauds. Statistical estimators combined with an in-depth analysis of the envelopes allowed us to identify 0.6% of fraudulent PNRs across 5 types of known frauds. An example of fraud carried out by travel agents is to make a booking without issuing payment, then take advantage of some booking engine functionalities to lock the booking for an unlimited period without observing the usual automatic cancellation of the ticket and price increase. Other fraud examples rely on flooding operations or abusive use of frequent flyer cards to be granted higher privileges.

Based on the hypothesis that the remaining PNRs are not fraudulent, we use this labelled dataset to evaluate the performances of the algorithms. Note that the computation of the precision and recall on this dataset will possibly exclude a few samples related to unknown frauds and must thus be adjusted when new frauds are discovered.

TABLE 1. DISTRIBUTION OF 5 AGGREGATED FEATURES

Feature	n_envelope	age_hours	n_segment	n_split	n_fop
mean	21.076	789.528	4.780	0.097	0.863
std	30.367	1326.969	22.846	0.599	1.400
min	1.000	0.000	0.000	0.000	0.000
5%	3.000	0.109	0.000	0.000	0.000
25%	6.000	45.538	2.000	0.000	0.000
50%	13.000	273.224	3.000	0.000	1.000
75%	26.000	856.738	5.000	0.000	1.000
95%	61.000	3547.940	12.000	1.000	3.000
max	1471.000	50258.785	2342.000	27.000	69.000

3. Outlier detection

Once the feature vector and sequence of actions of each PNR have been extracted and aggregated, outliers are detected according to the algorithms described in this Section.

3.1. Algorithms

The algorithms described hereafter cover several approaches of outlier detection though do not relate specifically to user behavior modelling since we focus on PNRs. A user-centric approach would be inappropriate to the extent that travel agents create and modify an important number of PNRs and each PNR can be modified by multiple agents. Half of the labelled dataset was used to perform a hyperparameter optimization for each algorithm targeting a maximum F1-score. The F1-score is the harmonic mean of precision and recall defined by $F_1 = 2 * \frac{precision*recall}{precision+recall}$. The best parameters were then used on the other half of the dataset to measure the precision, recall and F1-score for the testing phase. The performance measures written in this paper target the testing dataset in order to prevent overfitting. The following algorithms take the 82 numeric features as input, with the exception of HMM which uses the sequence of actions.

Probabilistic approaches considered here include the median absolute deviation (MAD) recommended for univariate outlier detection in [13]. It is a robust statistical measure computed as follows:

$$MAD = bM(|x_i - M|) \tag{1}$$

with $b = 1.4826$ a multiplicative constant used for normal distributions, x_i the values of an univariate dataset and M the median of the dataset. This measure can be used to identify outliers by thresholding the number of MAD between x_i and the median of the dataset as shown in:

$$\frac{x_i - M}{MAD} > \pm t \tag{2}$$

The overall score of a feature vector is then the maximum score obtained in each separate feature. Yet, this measure would flag as outliers any value different than the median if more than 50% of the data points have the same value, which has been observed for several features in our dataset.

This strong limitation prevent an efficient application of the MAD, which is why our benchmark used Z-score instead of MAD. This measure follows the same principles though is less robust since it relies on the average and standard deviation of the dataset μ and σ, performing a relaxed outlier detection in equation 3.

$$\frac{x_i - \mu}{\sigma} > \pm t \tag{3}$$

A Gaussian Mixture Model (GMM) has also been used to iteratively estimate the multivariate and multimodal distribution of the dataset where each feature is represented by a weighted sum of K normal distributions $P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$ where $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$ using an expectation-maximization (EM) algorithm to estimate the Gaussian parameters optimizing the log-likelihood. Thresholding the likelihood of a feature vector under the model allows a more efficient outlier detection than most algorithms according to the precision and recall described in Section 4.

Several distance based algorithms have been applied and raise the question of an efficient distance metric. The Euclidean and Mahalanobis distances have been used, the latter computing the distance between a given feature vector and the mean of the dataset normalized by the standard deviation of each dimension and adjusted for the covariance of those dimensions. DBSCAN is a robust parametric algorithm finding clusters of arbitrary shape in large datasets based on a density approach. Data points having less than m neighbors in a given radius are not clustered and thus flagged as outliers. This algorithm has been tested using Euclidean and Mahalanobis distances.

MeanShift is a clustering algorithm designed for datasets of smooth density and computes the centroids of clusters by optimizing the density function $f(x) = \sum_i K(x - x_i) = \sum_i k\left(\frac{||x-x_i||^2}{h^2}\right)$ with K a kernel function, x the initial estimate of the maximum of the density function, h a bandwidth given in parameter or estimated and x_i an input vector. A local maxima of this function is computed with gradient descent and results in cluster boundaries depending on the kernel function. This algorithm can be considered as a generalized expectation-maximization algorithm [2] and identifies outliers by selecting data points lying far away from the centroids of the clusters.

One-class SVM extends support vector machines (SVMs) by making use of unlabelled data to perform novelty detection. Standard SVM algorithms are binary classifiers finding a boundary between two classes by computing a linearly separating hyperplane in a high-dimensional space. The computations in high-dimensional space are achieved using kernel methods mapping points from the feature space to the high-dimensional space. One-class SVM fits a novelty detection boundary surrounding the training dataset by maximizing the margin between the dataset and the origin in the high-dimensional space. Overfitting is avoided by allowing a percentage of data points to fall outside the boundary using a regularization parameter ν which is an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors with $0 < \nu \leq 1$.

Hierarchical clustering builds a hierarchy of clusters according to an agglomerative or divisive approach, for example by minimizing the increase of total within-cluster variance after a merging step of the agglomerative process (Ward's method). Outliers can be further deduced by thresholding the outlyingness of the feature vectors computed according to the equations described in [17].

Regarding artificial neural networks, we implemented a Self-Organizing Map (SOM) mapping points from an input space to an output space. The output space is usually a 2-dimensional grid of neurons (Figure 3 after PCA processing for 2 components). Neurons are points in the feature space iteratively trained by being moved closer to dense regions

of data along with their neighbors. Outliers can be further identified according to [15] by thresholding the quantization errors (Figure 1), i.e. Euclidean distances between a point and the closest neuron. An additional step is required to detect outlying neurons attracted by clouds of outliers. Those are found by putting a threshold on the median interneuron distance (MID) matrix (Figure 2) containing the median of the distance between each neuron and its neighbors. The projections of known fraud samples on the nodes of the network are depicted by 5 distinct graphical artefacts in Figure 2. A distributed version of this algorithm showing similar results and convergence has been implemented based on the work of Lawrence et al. [12].

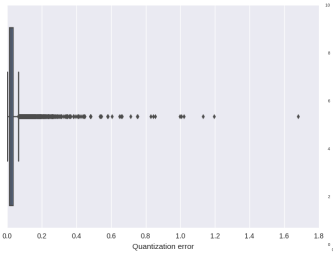

Figure 1. Box plot - Quantization errors

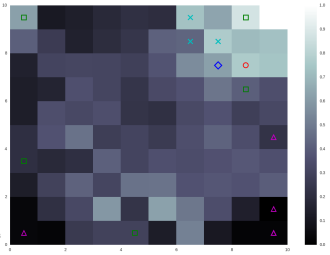

Figure 2. MID matrix - 10x10 SOM

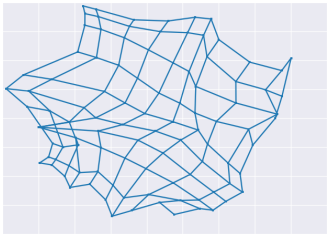

Figure 3. PCA representation of a 10x10 SOM - 40 iterations, $\sigma = 1.8$

As an additional set of experiments, we make use of the sequence of actions extracted and feed it into a Hidden Markov Model (HMM). This state-based approach models actions as observations which are generated by hidden states, each state having its own probability distribution. The system is then modelled as a Markov process describing the transition probabilities between the different states. Applying a threshold to the normalized log-likelihood of a sequence of actions under the model allows for outlier detection. The number of components used by the HMM is also found by hyperparameter optimization.

3.2. Multi-criteria decision analysis (MCDA)

The algorithms previously described output either a probability, a binary decision or a score which can be normalized according to an upper bound, e.g. by setting a maximum Z-score and assigning an outlier probability equal to 1 to all higher scores. Intuitively, aggregating the output of several algorithms will result in final scores conveying more confidence and leading to better performances than any single algorithm. To verify this assumption, we use Multi-Criteria Decision Analysis (MCDA) to design an aggregation model for the calculation of combined scores, taking as input all normalized scores given by individual algorithms.

One of the simplest MCDA operator is the weighted average $WA(x_1, .., x_n) = \sum_{i=1}^{n} w_i x_i$ with x_i a score and w_i a weight adjusted to the performance of a given algorithm, e.g. a F1-score. In the family of averaging functions, the Ordered Weighted Average (OWA) operator extends these functions by combining two characteristics: (i) a weighting vector (like in a classical weighted mean), and (ii) sorting the inputs (usually in descending order). OWA is a non-linear operator assigning a weight to each score based on its rank σ_i in the sorted list of scores (equation 4). OWA differs from a classical weighted means in that the weights are not associated with particular inputs, but rather with their *magnitude*. It can thus emphasize a subset of largest, smallest or mid-range values. In our application, the weights w_i are assigned based on a normal distribution such that the closer a score is to the middle one in the ranking, the higher the weight. The detailed method is described in [19]. Note that $x_{\sigma_i} <= x_{\sigma_{i+1}}$ and $\sum w_i = 1$.

$$OWA(x_1, .., x_n) = \sum_{i=1}^{n} w_i x_{\sigma_i} \tag{4}$$

As further generalization of OWA, Weighted Ordered Weighted Averaging (WOWA) [18] merges WA and OWA, by quantifying the reliability of the information sources (i.e. algorithms output in this case) with a vector *p*, and at the same time, allows to weight the values in relation to their relative ordering with a second vector *w* (as the OWA operator). It is formally defined as:

$$WOWA(x_1, .., x_n) = \sum_{i=1}^{n} x_{\sigma_i} \left[h(\sum_{j \leq i} p_{\sigma_j}) - h(\sum_{j \leq i-1} p_{\sigma_j}) \right] \tag{5}$$

where $h : [0, 1] \rightarrow [0, 1]$ is a non-decreasing function interpolating the points $(\frac{i}{n}, \sum_{j \leq i} w_j)$ together with the point $(0, 0)$. We can observe that if $p = (\frac{1}{n}, ..., \frac{1}{n})$ then WOWA returns the same result as OWA.

4. Results

Figure 4 shows the precision-recall curve obtained for a testing dataset containing 0.6% of known fraud samples. The highest F1-score reached on the plot is the weighted average of Z-score and self-organizing map. A precision below 100% is preferable since we are interested in unknown fraud discovery.

According to the results depicted, Euclidean distance ($minClustSize = 55, eps = 0.086$) provides better results than Mahalanobis distance ($minClustSize = 77, eps = 14.5$) when applied in DBSCAN. Though the latter has proven to be efficient in outlier detection performed on multivariate normal distributions, such hypothesis does not match most dimensions of our dataset. MeanShift ($seeds =$

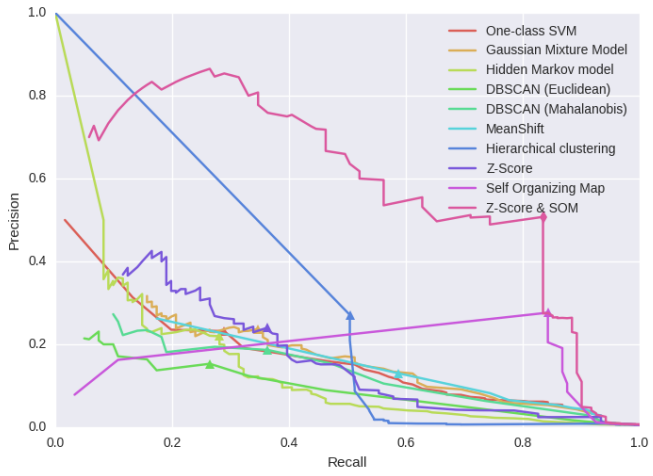

Figure 4. Precision-recall curves

$100, q = 0.94$) does not perform better than DBSCAN and both algorithms have to deal with high-dimensional data which affects the detection of high-density clusters. GMM (*2 components*) gives slightly better results than DBSCAN, separating the dataset into two clusters when DBSCAN only computed one.

Due to a dataset of complex distribution contaminated by outliers, one-class SVM ($kernel = RBF, \gamma = 0.07, \nu = 0.01$) is not able to model accurately a novelty boundary and does not perform much better than the previous algorithms, even though it makes an extensive use of the kernel trick to handle high dimensional data. Despite non-Gaussian distributions, Z-score ($t = 17$) is able to perform as well as one-class SVM due to the extreme values inherent to some fraud types such as flooding. Better results are also obtained from hierarchical clustering ($m = sizeDiff, t = 0.88$) though it requires the longest computation time. Since valuable information is lost by reordering the actions performed in each envelope, it prevents an efficient fraud detection from HMM ($states = 13, t = 0.67$). This algorithm also fails in detecting frauds relying on an important use of common actions when using a normalized log-likelihood.

The highest F1-score is reached with self-organizing maps ($neurons = 100, iterations = 40, qe_{Z-score_t} = 5.0, mid_t = 0.75$), reaching 28% precision for 84% recall. Figure 2 shows most fraud samples projecting on neurons having a high MID. Setting a MID threshold to 0.75 flags those neurons as outlying and allows a high recall while limiting the number of false positives.

All algorithms except HMM were able to detect much more easily the samples of 3 fraud classes among the 5 in the dataset. For those classes, some numeric features showed significant differences with the dataset average (e.g. flooding) or clear repetitions of actions at fixed intervals. HMM performed poorly on those frauds which involve a high number of common actions. However, it outperformed the other algorithms on a fraud based on unusual sequences of actions.

We remind here that DBSCAN and hierarchical clustering can only perform batch predictions for outlier detection and thus cannot compare a single PNR to an existing model, which may be an important limitation leading to model variations. MeanShift, SOM and one-class SVM return a binary output while the prediction step of GMM, Z-score and HMM compute an outlier score given an input vector. This score is turned into a ranking, hence making possible a fast insight on the most suspicious PNRs.

The benchmark of the aggregation operators has been performed with Z-score, hierarchical clustering and SOM. However, it does not allow us to assess the dominance of an operator over the others since OWA and WOWA require a higher number of values to demonstrate their efficiency. Nevertheless, excellent performances are achieved by aggregating the scores provided by Z-score and SOM, reaching 55% precision for 80% recall. Those two models perform fast computations on streaming data and predict the outlier score of up to 2,700 feature vectors per second and per thread. An in-depth study of the remaining unknown outliers revealed 3 new types of fraud and several misuses. The detailed results are given in Table 2.

TABLE 2. ALGORITHMS BENCHMARKING RESULTS OF PNR FRAUD DETECTION

Algorithm	Best F1-score	Precision	Recall
DBSCAN (Euclidean)	0.25	18.64	36.36
DBSCAN (Mahalanobis)	0.19	15.31	26.45
MeanShift	0.21	13.03	58.68
GMM	0.28	23.24	35.54
One-class SVM	0.26	23.49	28.93
Z-score	0.29	33.71	24.79
Hierarchical	0.35	27.11	50.41
HMM	0.26	24.44	27.27
SOM	0.42	27.64	84.30
Distributed Z-score	0.29	33.71	24.79
Distributed SOM	0.56	43.66	76.86
WA	0.55	39.78	88.43
OWA	0.57	76.39	45.45
WOWA	0.56	74.32	45.45
Z-score & SOM	0.63	50.75	83.47
Distributed Z-score & SOM	0.66	55.43	80.17

Since the best performances were achieved by aggregating Z-score and SOM, we implemented a distributed version of both algorithms. If Z-score is embarrassingly parallel, implementing a distributed SOM requires new equations to train the network on several chunks of data [12]. The speedup obtained by the distributed training of SOM is plotted in Figure 5 running in Scala on Spark with a hyperthreaded quad-core.

5. Conclusions and future work

We have performed an evaluation of 8 widely used machine learning algorithms and 3 MCDA operators in the context of unsupervised fraud detection applied to a particular dataset used in the global travel industry: Passenger Name Records (PNRs). Several distance metrics and outlier detection methods have been applied and result in a scalable and distributed model able to handle streaming

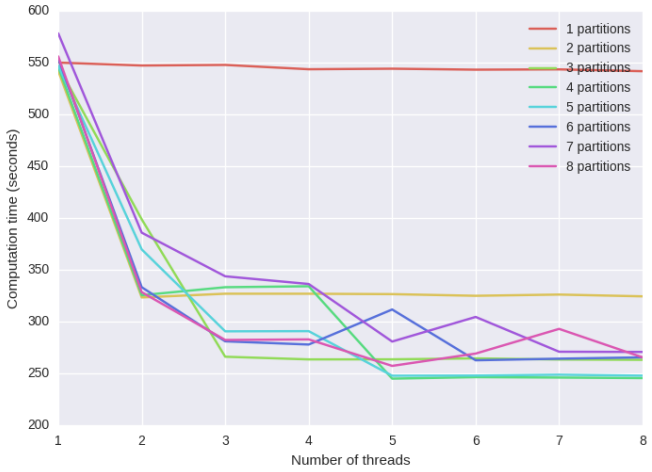

Figure 5. Computation time for the distributed SOM

data while performing an efficient fraud detection reaching currently 55% precision for 80% recall. Frauds detected by the final model can be ranked and provide a quick insight of the most critical situations. While we cannot disclose details on the specific frauds discovered in this study (for confidentiality reasons), the process described here has shown to be efficient for unsupervised fraud detection and can be applied to identify and prevent exploited flaws in Global Distribution Systems.

Future work includes an improvement of the system with a data enrichment step extending existing features by retrieving the location and time zone of the bookings based on airlines IATA codes, airports and travel agencies information.

We also consider a complementary evaluation focusing on the model robustness while removing the supervised optimization of the hyperparameters. This evaluation will aggregate the Z-scores of individual features according to the methods described in Section 3.2 to increase the precision of the algorithm and will use a reconstruction-based SOM to estimate the size of the neural network fitting the dataset. The number of components used by the GMM and HMM algorithms should be inferred by choosing a Dirichlet Process as a prior distribution on the number of clusters as described in [1].

Fed by the output of a preliminary supervised fraud detection process, such an ensemble unsupervised detection system will reinforce the overall fraud detection by returning previously undetected fraudulent samples to supervised algorithms in order to subsequently improve their detection capabilities. Faster manual fraud checks based on ranked outliers will be achieved by a visual analytics interface relying on box-plots and cross filters.

Acknowledgment

The authors would like to thank Prof. Maurizio Flippone for reviewing this paper and providing insightful comments.

References

[1] David M Blei, Michael I Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[2] Miguel A Carreira-Perpinan. Gaussian mean-shift is an em algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(5):767–776, 2007.

[3] David Andrew Clifton, Samuel Hugueny, and Lionel Tarassenko. Novelty detection with multivariate extreme value statistics. *Journal of signal processing systems*, 65(3):371–389, 2011.

[4] Paul A Crook, Stephen Marsland, Gillian Hayes, and Ulrich Nehmzow. A tale of two filters-on-line novelty detection. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3894–3899. IEEE, 2002.

[5] Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316, 1997.

[6] Johannes Gehrke, Venkatesh Ganti, Raghu Ramakrishnan, and Wei-Yin Loh. Boatoptimistic decision tree construction. In *ACM SIGMOD Record*, volume 28, pages 169–180. ACM, 1999.

[7] Sushmito Ghosh and Douglas L Reilly. Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE, 1994.

[8] Saikat Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE, 1999.

[9] Colin M-C Heilig P Colbath A Iron M, Zitkova M and Odgers M. Edifact implementation guide. *Passenger and airport data interchange standards*, 2013.

[10] Dongil Kim, Pilsung Kang, Sungzoon Cho, Hyoung-joo Lee, and Seungyong Doh. Machine learning-based novelty detection for faulty wafer detection in semiconductor manufacturing. *Expert Systems with Applications*, 39(4):4075–4083, 2012.

[11] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang. Constructing support vector machine ensemble. *Pattern recognition*, 36(12):2757–2767, 2003.

[12] Richard D. Lawrence, George S. Almasi, and Holly E. Rushmeier. A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. *Data Mining and Knowledge Discovery*, 3(2):171–195, 1999.

[13] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013.

[14] Geoffrey J McLachlan and Kaye E Basford. Mixture models. inference and applications to clustering. *Statistics: Textbooks and Monographs, New York: Dekker, 1988*, 1, 1988.

[15] Alberto Muñoz and Jorge Muruzábal. Self-organizing maps for outlier detection. *Neurocomputing*, 18(1):33–60, 1998.

[16] Iain Murray, David MacKay, and Ryan P Adams. The gaussian process density sampler. In *Advances in Neural Information Processing Systems*, pages 9–16, 2009.

[17] Luis Torgo and Maintainer Luis Torgo. Package dmwr. *Comprehensive R Archive Network*, 2013.

[18] Vicenç Torra. The weighted owa operator. *International Journal of Intelligent Systems*, 12(2):153–166, 1997.

[19] Zeshui Xu. An overview of methods for determining owa weights. *International journal of intelligent systems*, 20(8):843–865, 2005.

[20] Dit-Yan Yeung and Calvin Chow. Parzen-window network intrusion detectors. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 385–388. IEEE, 2002.

[21] Dit-Yan Yeung and Yuxin Ding. Host-based intrusion detection using dynamic and static behavioral models. *Pattern recognition*, 36(1):229–243, 2003.