Detecting Fraudulent Bookings of Online Travel Agencies with Unsupervised Machine Learning

Caleb Mensah[1], Jan Klein[2(✉)], Sandjai Bhulai[1], Mark Hoogendoorn[1], and Rob van der Mei[2]

[1] Vrije Universiteit, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
{s.bhulai,m.hoogendoorn}@vu.nl
[2] Centrum Wiskunde & Informatica,
Science Park 123, 1098 XG Amsterdam, The Netherlands
{j.g.klein,r.d.van.der.mei}@cwi.nl

Abstract. Online fraud poses a relatively new threat to the revenues of companies. A way to detect and prevent fraudulent behavior is with the use of specific machine learning (ML) techniques. These anomaly detection techniques have been thoroughly studied, but the level of employment is not as high. The airline industry suffers from fraud by parties such as online travel agencies (OTAs). These agencies are commissioned by an airline carrier to sell its travel tickets. Through policy violations, they can illegitimately claim some of the airline's revenue by offering cheaper fares to customers.

This research applies several anomaly detection techniques to detect fraudulent behavior by OTAs and assesses their strengths and weaknesses. Since the data is not labeled, it is not known whether fraud has actually occurred. Therefore, unsupervised ML is used. The contributions of this paper are, firstly, to show how to shape the online booking data and how to engineer new and relevant features. Secondly, this research includes a case study in which domain experts evaluate the detection performance of the considered ML methods by classifying a set of 75 bookings. According to the experts' analysis, the techniques are able to discover previously unknown fraudulent bookings, which will not have been found otherwise. This demonstrates that anomaly detection is a valuable tool for the airline industry to discover fraudulent behavior.

Keywords: Fraud · Anomaly detection · Unsupervised learning · Airline · Online travel agent

1 Introduction

Since industries have expanded their services to the internet to reach more customers, new ways have evolved to claim part of a company's revenue. Aviation faces a considerable problem with these malpractices. In 2008, airline industries

F. Wotawa et al. (Eds.): IEA/AIE 2019, LNAI 11606, pp. 334–346, 2019.
https://doi.org/10.1007/978-3-030-22999-3_30

all over the world missed out on 1.4 billion US dollars due to fraud. This was around 1.3% of their total revenue, although the rates were up to 4% in parts such as the Middle East and Latin America. Nowadays, these figures are expected to be even higher [1]. One of the conductors of fraud in the airline industry are online travel agencies (OTAs). Such an agency specializes in selling travel products including flights, hotels and rental cars to customers online. There is a wide variety of different kinds of OTAs, but they share at least one similarity: they all have an agency agreement with the supplier to resell its products [2]. In this case, the airline carrier allows the OTA access to its booking system to sell airplane seats. This expands the reach of the carrier, and therefore, increases its revenue. However, some OTAs violate the policies conducted by the airline organization in order to get access to cheaper ticket fares. This is possible, because an airplane seat can have a different price depending on several well-known factors. These include the seat's class (economy or business), the flight destination and the remaining time until departure. More specifically, when a flight consists of multiple flight segments, the price of a single segment can differ depending on the other segments in the complete flight. Here, a flight segment can be seen as the part between the departure and arrival of an airplane. If it lands more than once, there are multiple flight segments. An OTA can add one or more artificial segments to a flight to possibly get access to relatively lower prices. Later on, it can cancel these segments, which leads to revenue loss for the airline company. Therefore, the airline carrier desires to discover these malpractices to avoid losing profit.

In general, fraudulent behavior is assumed to be unusual, and hence, (largely) deviates from the expected, normal behavior. A way to discover such anomalous behavior is with the use of outlier detection techniques. Usually, the data with potentially fraudulent behavior is unlabeled, suggesting the use of unsupervised machine learning (unsupervised ML). This can be applied in a wide variety of domains, such as insurance, health care and cyber-security, with the same goal of finding malicious activities in data [3]. However, most of the applications are to discover and prevent bank fraud. For example, Bolton et al. propose the use of unsupervised profiling methods to detect credit card fraud in financial transactions on a customer-based level [4], while Ferdousi et al. examine the occurrence of fraud in stock market data as anomalous behavior in an evolving time series [5].

In the airline industry, the data consists of flight bookings, which can be seen as customer-based data changing through time. However, there are some important differences between bookings and financial transactions. First of all, customers are usually not aware of an OTA conducting fraud and are not directly affected by it. Fraud can even be advantageous to the customer who can purchase a cheaper flight ticket. Furthermore, a booking can be fraudulent because of how it changes through time, in contrast with fraud in a single financial transaction. Lastly, OTAs are part of the business model and are necessary for the airline carrier to make a profit. Of course, the majority of them act sincerely.

Since the airline industry has some characteristics which set it apart from other fields in which fraud occurs, it is interesting to examine how anomaly detection methods perform. More importantly, we were not able to find literature on the detection of fraudulent behavior of OTAs. This paper addresses that research gap. The contributions of our research are, firstly, to show how three different algorithms are applied to the booking data of OTAs to discover violations of the policies conducted by the airline carrier. This allows us not only to eventually block fraudulent bookings, but this can also enrich domain experts with new knowledge on how to avoid malicious behavior from happening. Before the techniques are applied, practically usable data is constructed from raw booking datasets. To this end, existing features are modified and new variables are added. Secondly, we show the importance of the engineered features in discovering fraudulent bookings. An evaluation set of 90 bookings is constructed for domain experts to classify as normal or fraudulent. We assess how well the anomaly detection methods are able to find these fraudulent observations.

2 Data

The data used in this research was obtained from an airline company. It has several kinds of features. The first type is based on the passengers' travel requirements information, summarized as a passenger booking. It consists of features such as travel dates, travel routes, ticket information and associated OTAs for all flights planned for the coming 360 days. There were some observations with missing values for some of the features. It was decided not to remove all of them, since having missing values in certain fields could be related to fraudulent behavior of OTAs. Missing information could be due to an error in the reservation system, which could have been exploited by an OTA. The second type of features contains information about revenue for each created booking. Actual revenue data is only available for ticketed (paid) reservations, while it is estimated for non-ticketed reservations using historical revenue data. The third type of variables is directed at the OTAs themselves. It provides characteristics such as an unique identifier and their location (or market).

The goal is to find fraudulent bookings and the corresponding OTAs which violate the policies of the airline carrier. The observations in this raw dataset were given on a flight segment level, but the data needs to be booking-based, i.e., each observation should indicate a booking. Therefore, the flight segments corresponding to the same booking had to be merged.

3 Methods

In this section, we discuss how the segment-based raw data was merged and how new variables were constructed from the raw features. Furthermore, we provide a preliminary analysis of the data and introduce which ML techniques were used for the experiments. Lastly, we discuss some transformations that were applied to the dataset with the goal to improve the results.

3.1 Feature Engineering

Before the experiments were carried out, new variables which better represent the underlying characteristics of the data were extracted from the raw features which were described in Sect. 2. They can be categorized into two classes: *(i)* revenue-based features, and *(ii)* booking-based features.

Revenue-Based Features. The first category of features was derived from the variables containing revenue information. These new features were introduced to describe the relative amount of revenue generated per booking and to compare the expected revenue with the ticketed revenue received. The predictions in revenue were based on a historical horizon of fifteen days, which was advised by domain experts. They expected the majority of the changes to occur during this time window. Moreover, the *predicted minimum* and *maximum* revenue were added as features and a feature describing the *changes in revenue* over the time horizon was included. A relatively large difference between the predicted maximum and actual revenue could indicate malicious booking behavior. Since these new features were obtained per flight segment, the records corresponding to the same booking were aggregated (by taking both the sum and average) to obtain one observation for each feature per booking. Furthermore, the *ticketing time* and a feature describing the *variation in ticketing times* for the flight segment were included. The ticketing time is the time it takes before a booking has been paid for. When a flight is legitimately booked online, the payment is expected to be done directly for the whole flight, and hence, there should be no or only a small variation in the ticketing times of the flight segments. A relatively large variation could indicate fraudulent behavior.

Booking-Based Features. The second class of features was composed from the raw booking features. These new features do not only describe the important characteristics of the booking, but they also represent the OTA providing the flight ticket. As mentioned before, flight segments corresponding to the same booking were aggregated to obtain one observation per booking. A new feature of interest is *point of commencement (PoC) circumvention*. This feature checks whether the effective PoC is equal to the true PoC. Here, the effective PoC is the starting point the passenger is expected to depart from, while the true PoC is the actual starting point. PoC circumvention occurs when a fake flight segment is added to a booking to get access to a cheaper fare. Before the airplane departs, this flight segment is canceled while the lower flight price is retained. A difference between the effective and true PoC of a booking coincides with one or more cancellations or additions of flight segments, so features were added which explicitly indicate this behavior. This was done by comparing booking data on successive days and by calculating the differences in the number of flight segments in each booking. Furthermore, variables which indicate whether an OTA chronologically books flight segments (from first departure until last arrival) were included. These features are directly linked to policy violations.

Lastly, several other features were composed which capture other booking related data, such as the *number of passengers* in a booking, the *length of stay*, *number of days between cancellations*, and so on.

Final Dataset. After the feature engineering process, the final dataset consists of $P = 84$ numerical features on $N = 17{,}886$ unique bookings. The total number of unique OTAs is 158. Now, anomalies can be found on a booking level. Each booking is connected to an OTA, making it possible to find the agencies which were potentially conducting fraud.

3.2 Feature Analysis

Before the ML algorithms were applied, a preliminary feature analysis on fifteen days of data was performed. The purpose of this study is to give an insight into the booking data in the considered airline market and to examine what fraudulent behavior of an OTA could be. The features of interest in this exploratory study were those which are concerned with PoC circumvention.

Table 1. Number of flight segments in six different bookings for the past fifteen days.

Booking	Days from now into the past														
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	7	7	7	7	7	7	7	7	7	6	6	6	6	6	6
2	6	6	7	7	7	7	7	7	7	7	7	6	6	6	6
3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6
4	7	7	7	7	7	7	7	7	7	7	7	7	7	6	6
5	2	2	2	2	2	2	2	2	2	2	4	4	2	2	2
6	6	6	7	7	6	6	6	6	6	7	7	6	6	6	6

To this end, the cancellations made in the bookings were examined. Table 1 shows the adjustments made in several bookings. These modifications were not just caused by cancellations, but also by the addition of new flight segments. This occurs in, for example, the last row. It shows an increase in the number of flight segments on day 12 and a decrease (cancellation) two days later on day 10, which is odd. This process repeats itself on day 5. It is unlikely that a passenger made such adjustments. A deeper analysis of a booking with canceled flight segments is shown in Table 2. Here, the two tables represent the same booking, but on different days. Note that the first two rows on the first day are not present on the second day anymore: these flight segments have been canceled. It is interesting to note that the values of the second column, the segment identifier, were not adjusted when flight segments were deleted. After examining this for several bookings with cancellations, it was concluded that the segment identifier

Table 2. Description of a particular booking on the first and second day. The columns indicate the departure date, segment identifier, departure location, arrival location, effective PoC, true PoC, and PoC circumvention, respectively.

Dep. day	Seg. ID	Dep. loc.	Arr. loc.	Eff. PoC	True PoC	PoC circumv.
Booking properties on day 1						
0	1	L_1	L_2	NA	PoC_1	NA
0	2	L_2	L_3	NA	PoC_1	NA
5	3	L_3	L_4	NA	PoC_1	NA
5	4	L_4	L_1	PoC_1	PoC_1	0
Booking properties on day 2						
5	3	L_3	L_4	NA	PoC_2	NA
5	4	L_4	L_1	PoC_1	PoC_2	1

Table 3. Overview of the descriptive statistics in the PoC features.

Description	Value
Percentage of PoC circumvented flight segments	7.27%
Number of unique effective PoCs	115
Number of unique true PoCs	138

was never modified. Hence, unexpected behavior in that variable could indicate this kind of fraud.

An overview of descriptive PoC characteristics is given in Table 3. Here, the percentage of flight segments with PoC circumvention is around 7%. Moreover, the table shows that the number of unique true POCs is greater than the number of effective PoCs. This difference indicates that there are at least 23 locations being used to circumvent the availability.

3.3 Anomaly Detection Techniques

Three anomaly detection techniques were considered in this research: isolation forest, one-class Support Vector Machine, and k-means clustering, which are explained in this section. These methods were chosen such that a wide variety of anomaly detection techniques was considered. Since no labeled data is available, unsupervised methods were used. They assume that the majority of the observations is normal, while only a small fraction is abnormal. This is the case for fraudulent bookings in the airline industry.

Isolation Forest. The first unsupervised technique was designed by Liu et al. in 2008. In contrast to traditional anomaly detection methods, an isolation forest explicitly separates anomalies rather than determining normal behavior and identifying anomalies as deviations from that behavior. This algorithm is

more effective and efficient in detecting anomalies than commonly used distance- and density-based methods [6]. In short, an isolation forest determines how long it takes for each observation to be separated, which is done by continuously splitting features between their minimum and maximum values. Since the splits are performed on a feature level, the importance of each feature can be easily derived. Each isolation tree $t \in \{1, \ldots, T\}$ in an isolation forest of $T \in \mathbb{N}$ trees yields a path length $h_t(\mathbf{x}_i)$ for every observation $\mathbf{x}_i \in \mathbb{R}^P$, $i \in \{1, \ldots, N\}$, with P the number of features and N the total number of observations. Anomalies are the records with the smallest average path lengths, because they can be isolated rapidly.

There are two hyperparameters in an isolation forest: the sub-sampling size $\psi \in \mathbb{N}$, and T. The first parameter controls the training data size per tree, while the second one determines how many isolation trees are constructed during training. The *anomaly score* $s_N(\mathbf{x}_i)$ determines how anomalous observation $\mathbf{x}_i$ is. It is defined as

$$s_N(\mathbf{x}_i) = 2^{-\frac{\overline{h}(\mathbf{x}_i)}{c(N)}} \in (0, 1),$$

where $\overline{h}(\mathbf{x}_i) = (1/T) \sum_{t=1}^{T} h_t(\mathbf{x}_i)$ is the average path length of $\mathbf{x}_i$ in the isolation forest and $c(N) = 2H_{N-1} - 2(N-1)/N$ is the expected path length with H_n the n-th harmonic number. Liu et al. offer some rules of thumb: if $s_N(\mathbf{x}_i) \gg 0.5$, then $\mathbf{x}_i$ can be considered as an anomaly; if $s_N(i) \ll 0.5$, then $\mathbf{x}_i$ can be regarded as normal; and if $s_N(\mathbf{x}_i) \approx 0.5$, then the status of $\mathbf{x}_i$ is vague.

One-Class Support Vector Machine. The second unsupervised method applied in this research was designed by Schölkopf et al. in 1999 [7]. The goal of this Support Vector Machine (SVM) is to identify one specific class amongst all observations. This results in trying to separate the observations belonging to the normal class from the rest of the feature space. Hence, the instances which do not lie within the non-linear normality boundary are considered to be anomalous. Therefore, one-class SVM (ocSVM) is a boundary-based algorithm. It has been shown that such algorithms perform better than density-based techniques, since they solve a fundamentally easier problem [8]. Consequently, ocSVM is widely used in the field of anomaly detection.

The goal of ocSVM is to separate the data from the origin with maximum margin. A mathematical problem (a quadratic program) is solved to determine the normality boundary, yielding an optimal normal vector $\mathbf{w}$ and margin ρ. There is a hyperparameter $\nu \in [0, 1]$ acting as a trade-off between the fraction of anomalies in the data and the number of training examples used as support vectors [9]. The anomaly score $s(\mathbf{x}_i)$ of an observation $\mathbf{x}_i$ is given by

$$s(\mathbf{x}_i) = \text{sgn}((\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}_i)) - \rho),$$

where $\mathbf{\Phi}$ is a map into a dot product space related to the chosen kernel function. Now, if $s(\mathbf{x}_i) < 0$, then $\mathbf{x}_i$ can be regarded as anomalous; if $s(\mathbf{x}_i) > 0$, then $\mathbf{x}_i$ can be considered normal; and if $s(\mathbf{x}_i) = 0$, then $\mathbf{x}_i$ is exactly on the boundary and its status is not determined.

***k*-Means Clustering.** The third and final anomaly detection technique considered is a clustering technique. k-means is an unsupervised, iterative algorithm proposed by Stuart Lloyd in 1957. It is one of the most popular clustering methods because of its simplicity [10]. In k-means, N observations have to be clustered into k clusters. Each cluster is represented by the mean (centroid) of the observations it contains. The clustering is performed such that the inter-cluster similarity is minimized, while the intra-cluster similarity is maximized. The similarity is determined by the Euclidean distance of the feature value to the mean value of the observations in the cluster: the smaller the distances, the higher the similarity.

The k-means algorithm converges quickly to a local optimum. Here, $k \in \mathbb{N}$ is a hyperparameter which, for example, can be determined using the elbow method. Here, the proportion of explained variance by the model is plotted as a function of the cluster size k. For small values of k, an increasing k will explain relatively much additional variance, but less additional variance is explained when k gets large. The optimal k is the value such that there is a bend in the plot. Now, to perform anomaly detection, a cluster boundary is introduced for each of the k clusters. This is a hypersphere around the cluster mean such that 95% of all the cluster observations are within the sphere, assuming that 5% of the observations are considered anomalous. For a new observation $\mathbf{x}_i$, first the closest cluster is chosen, and then it is determined whether $\mathbf{x}_i$ is within the boundary. If it is not, it can be considered anomalous.

3.4 Data Transformations

Finally, we investigated whether some data transformations had a positive effect on the anomaly detection performance of the algorithms discussed in Sect. 3.3. The transformations that were considered are normalization and standardization. To normalize the data, the feature values were linearly scaled such that all values lie in the interval $[0, 1]$. An advantage of normalization, or min-max scaling, is that each feature contributes equally, since all values are bounded in the same interval. Consequently, there is no feature overshadowing the other variables because of its large (absolute) values. However, a disadvantage is that the dispersion of the data is lost, possibly making it more difficult to detect anomalies. Standardization ensures that each feature has mean 0 and variance 1. The advantage of standardization over normalization is that the loss of dispersion is smaller.

Since tree-based models can handle varying feature ranges, normalization and standardization are not required in an isolation forest. However, the ocSVM and k-means methods are sensitive to magnitudes, and could therefore benefit from these transformations.

4 Experimental Setup

As mentioned in Sect. 3.3, there were several hyperparameters which had to be determined beforehand. For the isolation forest, these constants were the sub-

sampling size $\psi \in \mathbb{N}$ and the number of trees $T \in \mathbb{N}$. Liu et al. [11] argue that $\psi = 256$ and $T = 100$ are large enough to enable convergence of the average path length of each observation. Next, the parameter ν in the one-class Support Vector Machine was chosen to be 0.05, since we assumed that about 5% of the observations were anomalous. The number of clusters k, which was a hyperparameter for k-means, was determined by the elbow method and varied for the different experiments that were performed: $k = 2$ for no modifications, $k = 9$ for normalization and $k = 38$ for standardization. Since there are no labels, there was no ground truth in the data to which the hyperparameters could be optimized.

All described procedures were performed in Python 3.6 with the libraries numPy and Pandas. The results were obtained from an evaluation set. This set was determined by the anomaly scores calculated by the three discussed ML methods. For each technique, the anomaly scores of the bookings were ranked in a descending order. Then, a random subset of 10 bookings was taken from the top 30, one from the 30 scores around the median, and one from the 30 lowest scores, yielding a sample of 30 bookings for each anomaly detection method. The sample observations from the top 30 were predicted to be fraudulent, while the other observations were considered normal. In total, there were 3 samples of 30 bookings each. There was an overlap between the samples, i.e., the algorithms ranked some of the observations in the same regions. Hence, they were selected more than once for the samples. There were 75 unique bookings in the total of 90. The sample bookings were classified by the domain experts as fraudulent (1) or normal (0), making it possible to assess the detection power of the algorithms. In the samples 39 fraudulent bookings were found, while 36 bookings were deemed normal.

5 Results

5.1 Performance of Anomaly Detection Techniques

To determine the quality of the models, the precision, recall, F_1 score and F_2 score were calculated. The latter two are given by the formula

$$F_\beta = \frac{(1+\beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}},$$

Table 4. Results with no data transformation on method-specific evaluation samples

Model	Precision	Recall	F_1 score	F_2 score
isolation forest	0.75	0.8	0.774	0.789
ocSVM	0.769	0.588	0.667	0.617
k-means	1	0.444	0.615	0.5

where $\beta = 1$ for the F_1 score and $\beta = 2$ for the F_2 score. The F_2 score weighs recall more than the F_1 score does, i.e., it puts more emphasis on false negatives than false positives. This was done for the unmodified final dataset, which is the data without normalization or standardization. The performance metrics for the method-specific samples of size 30 are shown in Table 4. The construction of these samples is explained in Sect. 4. The isolation forest performed slightly better in finding policy violations (recall = 0.8) than making the distinction between normal and fraudulent observations (precision = 0.75). This was the other way around for the one-class Support Vector Machine (ocSVM) and k-means clustering, since both techniques had a higher precision than recall. Both the F_1 and F_2 scores suggest that the isolation forest performed the best. In fraud detection, reducing false negatives is usually more important than reducing false positives, since missing a fraudulent observation is deemed more harmful than raising a false alarm. False positives only bother domain experts with extra investigation time, while false negatives result in a potentially large loss of revenue. Hence, the F_2 score better represents how desirably the anomaly detection method performed. Note that this evaluation was done on the different method-specific samples. Although there is some overlap between them, a direct comparison of the performance measures is risky.

Table 5. Results with no data transformation on complete evaluation sample.

Model	Precision	Recall	F_1 score	F_2 score
isolation forest	0.706	0.615	0.657	0.632
ocSVM	0.75	0.462	0.571	0.5
k-means	0.8	0.308	0.444	0.351

The three ML methods were also applied to the complete evaluation sample. This set is the combination of the three method-specific samples of 30 observations each. The complete sample consists of 75 unique bookings. The results for the data with no modifications are presented in Table 5. The performance of the methods is comparable to the results on the method-specific samples in terms of F_1 and F_2 score. The isolation forest still ranks the best ($F_2 = 0.632$), followed by the ocSVM ($F_2 = 0.5$) and k-means clustering ($F_2 = 0.351$). This comparison was based on the same sample for each technique, thus strengthening the claims. Since there are 39 actual fraudulent bookings and 36 normal instances, the F_2 score is expected to be approximately 0.503 when the predicted labels are assigned by unbiased coin flips. This means only the isolation forest performed better than this threshold value.

The normalization and standardization procedures had remarkable influences on the results, as can be seen in Tables 6 and 7. For the method-specific samples, the precision and recall are both 0 for the isolation forest and ocSVM, performing severely worse than without data transformations. This was expected to some extent for the isolation forest, since the segregation of the observations is done

Table 6. Results on transformed method-specific evaluation data.

Model	Normalized				Standardized			
	Prec.	Recall	F_1	F_2	Prec.	Recall	F_1	F_2
isolation forest	0	0	0	0	1	0.067	0.125	0.082
ocSVM	0	0	0	0	0	0	0	0
k-means	0.909	0.556	0.690	0.602	0.8	0.444	0.571	0.488

Table 7. Results on transformed complete evaluation data.

Model	Normalized				Standardized			
	Prec.	Recall	F_1	F_2	Prec.	Recall	F_1	F_2
isolation forest	1	0.026	0.05	0.032	0.5	0.077	0.133	0.093
ocSVM	0.5	0.026	0.049	0.032	1	0.026	0.05	0.032
k-means	0.742	0.590	0.657	0.615	0.786	0.564	0.657	0.598

more rapidly with large variations in the data. However, this was not expected for the ocSVM. According to literature, transforming the data should benefit an SVM. This could be due to the Gaussian radial basis function that we used in this research. Nevertheless, the performance of k-means clustering increased from $F_2 = 0.5$ to $F_2 = 0.602$ with normalization. There was a slight decrease for standardization from $F_2 = 0.5$ to $F_2 = 0.488$.

For the combined sample, the performance of all three considered anomaly detection methods moderately increased in terms of F_2 score compared to the method-specific samples. In short, the isolation forest performed the best on its own sample of 30 observations without any data transformations ($F_2 = 0.789$). This was also the case for the one-class SVM ($F_2 = 0.617$). k-Means clustering performed the best on the complete evaluation sample with normalized data ($F_2 = 0.615$). Note that all these values are larger than the threshold value of 0.503.

5.2 Feature Evaluation

The results of the anomaly detection methods and the advise of the domain experts allowed us to construct a set of features which were deemed to be the most likely to identify suspicious behavior of an OTA. The list of the five most important features is given in Table 8. The first feature indicates the sum of the segment identifiers divided by the corresponding triangular number: $\binom{S+1}{2}$, where S is the number of flight segments in the booking. We expect the sum to equal the triangular number (and so the feature value to be 1), since the segments are usually labeled in an ascending order from 1 to S. The order ratio feature is not equal to 1 for the booking shown in Table 2, because flight segments have been canceled. The second variable is related to PoC circumvention.

As discussed in Sect. 3.2, the fact that PoC circumvention has occurred could indicate fraudulent behavior. We also showed in Tables 1 and 2 how the number of cancellations, which is the third most important feature, could be connected to fraud. The fourth feature has not been discussed in the feature analysis, but an unexpected value of this feature also suggests malicious behavior. Finally, the last feature in Table 8 indicates whether the OTA creating the booking is not equal to the OTA owning it.

Table 8. List of features to identify suspicious activity.

List of features that detect suspicious activity
Order ratio
PoC circumvention ratio
Number of cancellations
Number of booking class switches
Number of OTA owners which are unequal to the creator

6 Discussion and Conclusion

The goal of this research was to discover policy violations conducted by OTAs with the use of three anomaly detection methods. To this end, the raw data was analyzed and new variables were constructed to better describe the behavior of the OTAs. We demonstrated that these new features were important in detecting fraudulent bookings. This encourages domain experts to monitor these variables to detect some of the fraudulent behavior and avoid revenue loss. Moreover, this advises an airline organization to update its policy agreement with the OTAs to prevent such malpractices from happening in the future.

Together with the domain experts, we concluded that most of the anomalies were caused by cancellation activity in the bookings, suggesting that the values of the features corresponding to this behavior give a strong indication of fraud. However, there were instances in which normal bookings were detected as fraudulent, which happened because of complex and highly unusual flights. Also, there were instances in which the domain experts marked a booking as fraudulent, but it was based on a gut feeling. Here, the benefit of using unsupervised ML becomes evident: these bookings would never have been found when the bookings were only analyzed on a feature-based level. Moreover, since we were not able to find literature about this research field, we took an important step in understanding fraudulent behavior conducted by OTAs.

One of our suggestions for future research is to broaden the scope to make the results more generalizable. Firstly, we considered the bookings of one airline market. It is possible that the behavior of OTAs is significantly different for another market. Secondly, because of time constraints, only 75 records ($\approx$0.42%)

were analyzed by the domain experts. This means the results could be notably different when a new sample is considered. Another suggestion for future research is to find out at which stage in the booking process the models are able to detect fraud in an online setting.

References

1. Centre for Aviation: Fraud costs airlines USD1.4 billion a year. Regional airlines the fraudsters' "carriers of choice". https://centreforaviation.com/analysis/reports/fraud-costs-airlines-usd14-billion-a-year-regional-airlines-the-fraudsters-carriers-of-choice-48150. Accessed 2 Apr 2019
2. Rezgo Booking Software: What is an OTA? https://www.rezgo.com/glossary/ota. Accessed 2 Apr 2019
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), Article No. 15 (2009)
4. Bolton, R., Hand, D.: Unsupervised profiling methods for fraud detection. Credit Scoring Credit Control **7**, 235–255 (2001)
5. Ferdousi, Z., Maeda, A.: Unsupervised outlier detection in time series data. In: 22nd International Conference on Data Engineering Workshops, pp. 51–56. IEEE, Atlanta (2006)
6. Liu, F., Ting, K., Zhou, Z.-H.: Isolation forest. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 413–422. IEEE Computer Society, Washington (2008)
7. Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J.: Support vector method for novely detection. In: Advances in Neural Information Processing Systems, pp. 582–588 (2000)
8. Tax, D., Duin, R.: Uniform object generation for optimizing one-class classifiers. J. Mach. Learn. Res. **2**, 155–173 (2001)
9. Heller, K., Svore, K., Keromytis, A., Stolfo, S.: One class support vector machines for detecting anomalous windows registry accesses. In: Proceedings of the workshop on Data Mining for Computer Security, vol. 9. IEEE, Melbourne (2003)
10. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theor. **28**(2), 129–137 (1982)
11. Liu, F., Ting, K., Zhou, Z.-H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data (TKDD) **6**(1), 4 (2012)