Predictive Analytics to Prevent Voice over IP International Revenue Sharing Fraud

Yoram J. Meijaard, Bram Cappers, Josh Mengerink, Nicola Zannone

Predictive Analytics to Prevent Voice Over IP International Revenue Sharing Fraud

Y.J. Meijaard[1], B.C.M. Cappers[1 2], J.G.M. Mengerink[2], and N. Zannone[1]

[1] Eindhoven University of Technology, Netherlands
y.j.meijaard@student.tue.nl, b.c.m.cappers@tue.nl, n.zannone@tue.nl
[2] AnalyzeData, Netherlands
b.cappers@analyzedata.com, j.mengerink@analyzedata.com

Abstract. International Revenue Sharing Fraud (IRSF) is the most persistent type of fraud in the telco industry. Hackers try to gain access to an operator's network in order to make expensive unauthorized phone calls on behalf of someone else. This results in massive phone bills that victims have to pay while number owners earn the money. Current anti-fraud solutions enable the detection of IRSF afterwards by detecting deviations in the overall caller's expenses and block phone devices to prevent attack escalation. These solutions suffer from two main drawbacks: *(i)* they act only when financial damage is done and *(ii)* they offer no protection against future attacks. In this paper, we demonstrate how unsupervised machine learning can be used to discover fraudulent calls at the moment of their establishment, thereby preventing IRSF from happening. Specifically, we investigate the use of Isolation Forests for the detection of frauds before calls are initiated and compare the results to an existing industrial post-mortem anti-fraud solution.

1 Introduction

International Revenue Sharing Fraud (also known as Toll fraud) is a multi-billion dollar scheme [17] where fraudsters use telecommunications products or services without the intent to pay. With Voice Over IP (VoIP) enabling devices to communicate over the Internet, VoIP devices and services are a popular target for hackers to abuse. The result is typically phone bills of thousands of euros [28] that victims have to pay. In practice, network operators often covers these expenses either because they were responsible for the vulnerability or they want to maintain customer satisfaction. To minimize this fraud, they often install anti-fraud detection software to detect signs of fraud as quickly as possible. Current anti-fraud solutions typically use pricing information and features such as call duration to determine whether a call was fraudulent. Although such detection turns out to be accurate [4], those features are only available after the call has been made. Since the detection of a fraudulent call does not provide any information about the severity of the next call, operators are often forced to shutdown the phone device or underlying VoIP server (e.g., a Private Branch eXchange) until they are sure that the call was legitimate. The result is that customers can no longer make (regular) phone calls until the fraud has been resolved.

The goal of this work is to explore to what extent fraud can be detected when a call is established, compared to existing industrial (post-mortem) anti-fraud solutions. To this end, we study how we can use unsupervised machine learning solutions and, in particular, Isolation Forests [25] to determine the severity of a phone call using features that are only available at the start of a call. In contrast to traditional anomaly detection algorithms (see [8] for a survey), Isolation Forests enable the detection of anomalies without having to construct profiles of normal behavior. This makes the algorithm particularly suitable for highly dynamic environments where construction of normal profiles can be too resource intensive. In addition, with linear time classification Isolation Forests have also shown to be suitable for the analysis of large data streams [9]. However, in order for machine learning solutions to be applied for fraud detection, the number of false alarms need to be kept to a minimum. This typically requires properly tuning such solutions and underlying parameters, which is a non-trivial task [16].

In this work, we perform a latitudinal study to assess the detection capabilities of Isolation Forests in the context of IRSF detection and to study how different parameter settings and feature spaces affect performance. Specifically, our main contributions are:

- the application of Isolation Forest anomaly detection for the early-stage detection of IRSF;
- a case study of the approach demonstrating the effect of different parameter settings, feature sets, and the use of derived features to improve detection rates;
- a comparison of the detection with respect to existing post-mortem analysis;
- a number of lessons learned on how to use Isolation Forests for the detection of anomalies in multivariate data.

Our approach offers several benefits compared to existing anti-fraud solutions. In particular, it enables network operators to detect signs of fraudulent calls before they are established, providing operators the opportunity to block the call preemptively rather than blocking a phone device entirely when the fraud has happened.

The remainder of the paper is structured as follows. The next section introduces background on VoIP and related frauds. Section 3 discusses related work. Section 4 presents our methodology and Section 5 presents its experimental evaluation. Finally, Section 6 discusses the results along with the limitations of the approach and Section 7 concludes the paper and provides directions for future work.

2 Background & Motivations

Voice over IP (VoIP) enables users to communicate audio and video over the Internet. Compared to physical phone lines, the Internet provides a cheaper alternative and is widely available. The VoIP infrastructure consists of four main types of components: VoIP devices, Private Branch eXchanges (PBX's), the Public Switched Telephony Network (PSTN), i.e. the legacy phone infrastructure, and VoIP gateways. A VoIP device connects over the Internet to a PBX operated by a network operator, which in turn connects to other PBX's. Over this network, a VoIP device is able to call other VoIP devices. VoIP gateways connect the VoIP network to the PSTN to enable VoIP devices to call legacy phones.

Over the years, several vulnerabilities in VoIP and underpinning network protocols have been discovered. Sahin et al. [32] define a taxonomy of VoIP fraud schemes in which these schemes are categorized with respect to their root causes along with their weaknesses and techniques to exploit them. Given that IRSF has been recognized as the largest class of frauds in practice [17], in this work we focus on the detection of this class.

In IRSF an attacker breaks into the VoIP system of the victim and places numerous calls to a premium phone number, i.e. a number that charges a fee in addition to the regular cost of the call, which is owned by the attacker or a colluding entity. These calls are charged to the victim and the revenue made from this call is shared amongst the attackers. Blacklisting premium numbers is virtually impossible, due to them being ill-defined internationally [32]. Therefore, network operators need methods for the detection of IRSF. Ideally, these methods should be able to detect the fraud when the call is established to prevent it from happening.

Real-time anomaly detection in telco industry, however, is challenging due to the variety and volumes of data [12]. Depending on the type of users, the number of calls can vary from tens to hundreds per month. In addition, data volumes are often too large to analyze all network packets individually. As a result the use of profiling techniques can be too computational intensive.

In order for anomaly detection to be effective in this field, the evaluation of a data point needs to be reliable and efficient. False positives need to be kept to a minimum to avoid operators from being overloaded with false alarms. In particular, the maximum allowable false positive rate should be $<2\%$ [23]. The false negative ratio (i.e., the number of missed fraud) is less critical here. We wish to catch as much fraud beforehand as possible, but missed fraud cases could still be detected by a post-mortem detector. In addition, some features such as the call duration and cost are only available after the call has been made. Most data fields in VoIP calls consists of categorical features. In order to enable real-time detection, the evaluation of a new data point and model updating must be efficient [1].

In summary, in order to make the detection of IRSF during call establishment effective, the anomaly detection method should meet several requirements. Specifically, a solution for *online* IRSF detection should:

R1 Be resilient to ill-balanced label distribution (e.g., 99% normal and 1% fraud).
R2 Support the analysis of categorical features.
R3 Enable fast classification of (new) individual calls.
R4 Provide a computational efficient method for training the classification model.
R5 Generate an operationally feasible number of false positives.
R6 Not rely on call features that are only available after the end of a call.

In the next sections we discuss to what extent existing solutions meet the requirements and show how Isolation Forests can be used to solve the detection task.

3 Related Work

Fraud detection is an extensively studied field covering a wide variety of techniques [22]. We first give a broad overview on data analysis techniques used to detect IRSF, followed by a detailed discussion on existing anti-fraud solutions that are based on the analysis of Call Detail Record (CDR) logs. An evaluation of existing solutions with respect to the requirements identified in Section 2 is given in Table 1. For a more detailed overview on the use of anomaly detection techniques for the detection of telecom fraud, we refer to [19].

Table 1. Comparison of fraud detection techniques with respect to the requirements for online IRSF detection, as formulated in Section 2. In the table, ● means "support", ◐ "partially support", ○ "no support".

Requirements	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**
Wiens et al. [38]	●	●	◐	○	●	○
Kuble [23]	●	●	◐	○	●	○
Olszewski [27]	●	◐	●	○	○	○
Wiens et al. [37]	●	●	◐	○	●	○
Becker et al. [3]	●	●	○	○	○	○

Fraud Detection. Similar to intrusion detection [13], approaches for the detection of (IRSF) fraud can be classified in two main categories, namely *signature-based* detection and *behavioral-based* detection [5]. Signature-based detection uses fixed rules to identify whether a call adheres to predefined patterns. For instance, a suddenly increase in a user's dial expenses compared to his average phone bill provides an indicator for IRSF. For more details on the typical signatures used for fraud detection, we refer to [14]. Although the use of signatures provides an effective approach to fraud detection, it is unable to discover new patterns enabling the detection of IRSF when, for instance, the cost of the call is unknown.

These shortcomings are typically addressed by *behavioral-based* detection techniques. These techniques focus on the (statistical) profiling of entities so that the regularity of new data points can be determined according to a model of expectation derived from historical data. This enables analysts to automatically derive models that are tailored on a per profile basis. In case of IRSF detection, we can identify two main types of behavioral-based techniques, namely *offline* and *online* [7, 36]. *Offline* behavioral-based techniques (also referred to as *post-mortem* detection) stores incoming data and do the analysis after all calls have been made. *Online* techniques (a.k.a. (near)-real time analysis) perform the analysis the moment new data arrives (e.g., during call establishment).

Our work can be positioned as online behavioral-based method. In particular, we investigate the possibilities of applying online anomaly detection at the start of a call, enabling the early stage detection of IRSF. This is in contrast to current literature, which mainly focuses on the offline analysis of calls by applying anomaly detection on Call Detail Record (CDR) logs [29]. Next, we review those solutions.

CDR-based fraud analysis. A common approach in the telecommunication domain for offline behavioral-based fraud detection is through the analysis of Call Detail Records (CDR). For instance, Modani et al. [26] use decision trees and logistic regression to predict the churn rate of companies in CDR records whereas Wiens et al. [37, 38] apply statistical profiling on user call behavior to detect exploited FRITZ!Boxes in a network. The use of anomaly detection techniques are also shown to be useful for the discovery of unknown patterns in call records. For instance, Becker et al. [3] are one of the first to use unsupervised learning on CDRs to discover cellphone usage patterns. CDRs have also been used to discover relationships between criminals based on the assumption that criminals interact with each other. Specifically, Kumar and colleagues propose a model to construct a CDR database in which these relationships are captured [24].

Kübler et al. [23] analyze toll fraud hindsight by clustering user behavior using unsupervised learning techniques such as k-means clustering and EM mixture models [2]. Instead of using cost-based features, they build user-profiles based on destination number and duration of the call. Although this approach to discover behavioral patterns through feature engineering is similar to ours, it requires the construction of user profiles to determine the severity of a call. In addition, the clustering algorithms proposed in [23] are too computational intensive to be used in online settings. Especially in dynamic environments where users register and leave phone operators on a regular basis, building "normal" profiles can be too resource intensive to be applied in practice. The advantage of using Isolation Forests is that this technique does not require a normal baseline to determine the severity of an anomaly, but aims to isolate "few and different" points from the rest of the data [25].

In summary, current approaches for fraud detection are signature-based or rely on an offline behavioral-based analysis of calls using CDR logs. The absence of offline features such as cost and duration in online detection requires adaptation of existing fraud detection techniques both in terms of the feature space to analyze as well as computational requirements. Our work overcomes these limitations by using Isolation Forests.

4 Methodology

The goal of the methodology is to explore to what extent we are able to detect fraudulent IRSF calls before they are established, giving operators the opportunity to block the call and prevent the fraud from happening.

We enable online IRSF detection through a real-time analysis of the network traffic from and to Private Branch eXchange servers to identify suspicious patterns in the establishment of VoIP calls. The establishment of a VoIP call is achieved by means of a handshake using the Session Initiation Protocol (SIP) [31]. This protocol contains call meta-data similar to Call Detail Records such as call start time, source and destination numbers, and user identifier. In addition, more advanced features can be derived such as the country of origin or in/outside office hours (we refer to Section 5.4 for more details). Since the analysis is done during call establishment, we assume that features such as call duration, cost of a call and call end time are unavailable.

Our methodology for online detection of IRSF comprises three steps, as depicted at the top of Figure 1:

1. From the VoIP data we extract the features to use for training of the isolation forest.
2. We generate an isolation forest from the training data, which assigns an anomaly score to each call.
3. We classify all calls with an anomaly score lower than a given threshold as anomalies. Anomalous records are marked as `fraud` while the other are marked as `normal`.

To validate experimental results, we use an industry post-mortem fraud detector as a baseline. Validation comprises three steps:

5. After the call ended, a CDR record is created by the underpinning PBX server.
6. The industrial anti-fraud detector labels the call records either as `fraud` or `normal`.

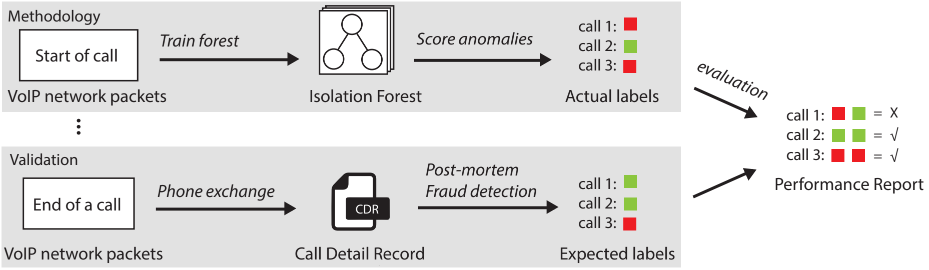

Fig. 1. At the start of a call the isolation forest is trained and anomaly scores are computed on the data points. At the end of a call, the post-mortem anti-fraud detector evaluates the call after which the labels of the isolation forest are compared to the existing solution.

7. The evaluation phase compares the labels from our approach with the ones given by the industrial detector and generates a performance report.

In the remainder of the section, we first introduce Isolation Forests after which we discuss the dataset used for our experiments.

4.1 Isolation Forests

Traditional anomaly detection techniques typically construct a profile of the data that is considered “normal“ and evaluate any new data point against that model [8]. Isolation Forests use a different approach as it aims to separate anomalous data points in the dataset without building a distribution or profile. Instead, data is sub-sampled in a tree structure by evaluating data points on randomly chosen features and split on those features. Specifically, if a node contains two or more records, it is split according to a randomly chosen feature. This causes an anomalous data point to reside in their own leaf node. This is also illustrated in Figure 2. The main idea is that similar data points require more splits before they can be separated from one another while anomalies remain close to the root. Since purely random splits can lead to poor decision trees, data points are evaluated against a collection of generated trees to determine whether they are statistically significantly different from the rest.

The anomaly score of a data point is based on its average depth in all isolation trees. A high score indicates that many splits are required in order to separate a data point from the rest and is therefore considered to be similar to other records. Similarly, if the score is low, only few splits are required to separate the data point and, therefore, it is considered different from the other data points. A user defined threshold is used to determine whether a score is low or high, i.e. whether a data point should be considered anomalous.

In our work, we evaluate the following parameters for the generation of Isolation Forests (illustrated in Figure 3):

– `nrTrees` represents the number of trees in the forest.
– `sample_size` represents the size of the sample set. The sample size determines the number of records for the construction of a tree in the forest.
– `max_depth` represents the maximum depth for each isolation tree.

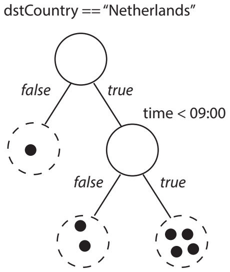

Fig. 2. Example model where seven data points are clustered according to two randomly chosen data splits. Regular data points are assumed to have overlap in values and are therefore harder to separate from one another compared to anomalous data points. Isolation forest uses the path length from the root to determine data points' regularity. Anomalous data points have their own leaf node.

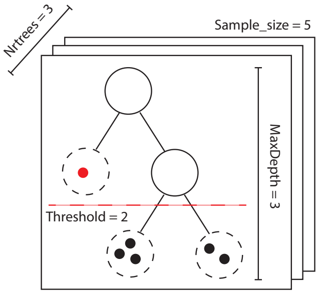

Fig. 3. Parameters in isolation forests manipulate the forest by specifying the number of trees, number of nodes per tree, the maximum depth of generated trees along with the number of samples that must be used for the construction of a tree. The dot in red is anomalous with respect to the chosen threshold.

– `threshold` is a user-defined threshold that determines the average tree depth that should be exceeded in order for a data point to be considered `normal`.
– `feature_set` represents the set of features used for the construction of the trees in the forest.

Isolation Forests provide several advantages compared to other machine learning techniques. Compared to black-box techniques such as neural networks, Isolation Forests are transparent. By inspecting the different trees, a VoIP operator is able to provide an explanation on why a data point was considered anomalous, thereby enabling means to judge the quality of a model using domain knowledge. Adaptations of Isolation Forests have been proposed to make them suitable for streaming data [33]. In contrast to other unsupervised learning methods such as k-means and EM-Mixture models [2], the construction of the forest is computationally efficient, since distances between data points are not computed, but features and data splits for the trees are chosen randomly. In addition, the classification of a new data point requires at most $\mathcal{O}(\texttt{max\_depth})$ steps per tree and can be run in parallel for every tree in the forest. Compared to supervised learning techniques such as decision tree classifiers and SVM, Isolation forest do not require labels to identify anomalous data points.

4.2 Data characteristics

We performed a number of experiments to assess the effectiveness of Isolation Forests for the online detection of IRSF. Our dataset consists of over 10.000 VoIP calls, out of which the industrial anti-fraud solution marked 150 calls as fraudulent. The dataset consists of nine Dutch users where a user can represent a physical person, a server, or an entire phone operator. The data was recorded for a month by mirroring network traffic from a

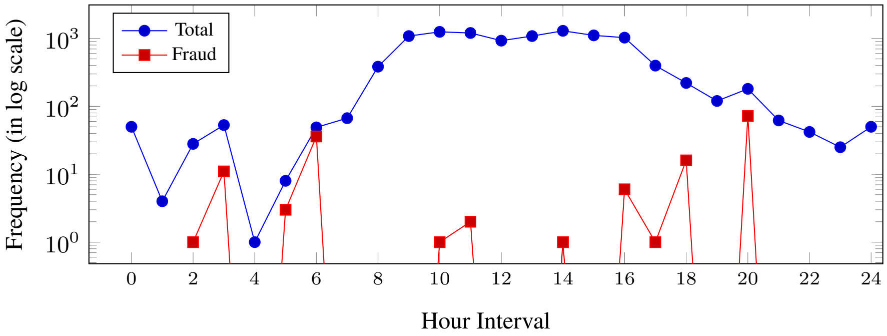

Fig. 4. Distribution of calls over the time of the day in our dataset.

Private Branch eXchange server of a Dutch VoIP provider. Each call is characterized by the following online (i.e., pre-call) features:

- `record_id` is the identifier of the VoIP record.
- `user_id` is the identifier of the user that made the call.
- `srcNr` is an anonymized representation of the source number.
- `srcCtry` represents the country of call source.
- `dstNr` is an anonymized representation of the destination number.
- `dstCtry` represents the country of the call destination.
- `disposition` indicates whether the call was answered, canceled or busy.
- `time` is a timestamp indicating when call took place.

To test the suitability of Isolation Forests for the detection of IRSF in general, we also collected the following offline (i.e., post-call) features:

- `duration` indicates how long the call lasted in seconds.
- `billsec` indicates the cost of the call per second.

We performed an analysis of the dataset to verify that it does not contain any artefacts that could disqualify the experiments. For the sake of space, here we only report our analysis with respect to features `time` and `dstCtry`. As shown in Figure 4, calls are typically made during work hours whereas fraud cases mainly occur outside of work hours. We can observe in Figure 5 that the destination of the calls is predominantly to the Netherlands, but there is a wide variety of calls to different countries. The plot also shows that simply blocking calls to certain countries is not a viable solution to fraud prevention, as it would incorrectly block at least as much normal traffic.

5 Experiments

This work aims to evaluate the effectiveness of Isolation Forests for the detection of IRSF. In particular, we are interested in answering the following research questions:

Q1 Are Isolation Forests suitable for offline and online detection of IRSF ?
Q2 What is the effect of different parameter settings towards detection rates?

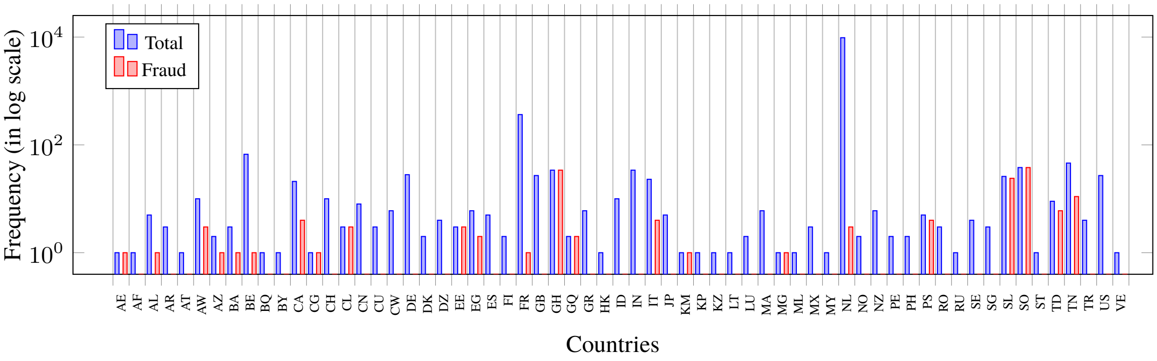

Fig. 5. Distribution of calls over countries (expressed using ISO 3166-1 country codes) in our dataset.

Q3 What is the effect of different feature sets towards detection rates?
Q4 To what extent does the use of derived features improve detection rates?

Questions **Q1** and **Q2** validate whether Isolation Forests can be used in general for IRSF detection. **Q3** aims to determine which features in the input data should be taken into account during classification. The naïve usage of features such as `time` or categorical features can significantly blow up the state space in which the algorithm needs to operate and can lead to sub optimal results [10, 20]. Similar to Kübler et al. [23], in **Q4** we test the effect of discretizing the `time` feature to see how this influences detection rates.

For the implementation of Isolation Forests we used the h2o framework (version 3.26.0.10) [6]. This framework offers out-of-the-box support for categorical data and is commonly used in academia to study machine learning techniques.

Settings. In our experiments, we tested different configurations for the generation of Isolation Forests by varying the parameters presented in Section 4.1. An overview of the parameters for each experiment is given in Table 2. It is worth noting that increasing `nrTrees` does not influence the outcome of the model, since it is statistically unlikely to generate significantly different anomaly scores when averaging over 100 trees given the feature set of VoIP calls. Therefore, the recommended default settings for this parameter is used, i.e. 100. In Experiments 1, 3, and 4 we set parameter `sample_size` to 256, which we deem sufficiently large. This choice is further motivated by Experiment 2 in which we varied this parameter to study its effect on detection rates.

The choice for parameters `max_depth` and `threshold` is based on two observations: Isolation Forests are unable to classify anomalies from normal behaviour with a small `max_depth`, e.g., ≤ 5. On the other hand, a large `max_depth` (e.g., ≥ 15) can result in problematic overfitting, as mentioned in the original paper [25]. Simultaneously, the value of `threshold` cannot exceed the `max_depth` value and, therefore, its domain is restricted. Specifically, we varied `threshold` in the range [`5,max_depth`], with steps of 0.1.

Evaluation metrics. In general, the evaluation of an unsupervised learning algorithm is a challenge due to the lack of labeled data or a proper baseline. Inspired by Wang et

Table 2. Summary of parameter settings used for the experiments described in Section 5.

Parameter	Experiment 1	Experiment 2	Experiment 3	Experiment 4
`nrTrees`	100	100	100	100
`sample_size`	256	$64, 128, 256, 512$	256	256
`max_depth`	$[6, 15] \in \mathbb{N}$	$[6, 15] \in \mathbb{N}$	$[6, 15] \in \mathbb{N}$	$[6, 15] \in \mathbb{N}$
`threshold`	$[5, 15] \in \mathbb{R}$	$[5, 15] \in \mathbb{R}$	$[5, 15] \in \mathbb{R}$	$[5, 15] \in \mathbb{R}$
`feature_set`	pre-call, post-call	pre-call	subsets of pre-call	pre-call + derived features

al. [35] and Dudoit et al. [11], we measure the quality of the resulting model by means of an external index using the labels of the existing anti-fraud solution as the baseline.

The most common evaluation metric to assess the performance of a machine learning algorithm is the Receiver Operating Characteristics (ROC) curve [18]. In this curve the true positive rates and false positive rates are plotted for an algorithm at various thresholds (i.e., the Isolation Forest `threshold` parameter). In our application domain, the goal is to have a curve where the true positive rate is high and false positive rate is close to 0. The Area Under the Curve (AUC) is an indicator how well the algorithm can discriminate between normal and fraudulent traffic. The AUC can vary between 0.0 and 1.0 and the larger the area, the better the performance.

5.1 Experiment 1: Effectiveness of Isolation Forests for IRSF detection

In this experiment, we assess whether the classification capabilities of Isolation Forests are suited to fraud detection in VoIP. We test this by applying Isolation Forests to both an online and an offline setting. Specifically, we conduct two tests: one using all data available at the establishment of the call (e.g., pre-call features) and one also including post-call features (cf. Section 4.2). Based on **R5**, we require that the false positive rate is less than 2%, while the true positive rate should be sufficient to outweigh operational costs.

The full construction of an isolation forest for the dataset is efficient and takes approximately 1 second (**R4**). Figure 6 reports the ROC curves when using pre-call and post-call features. The figure shows that, at a 2% false positive rate, the true positive rate obtained using pre-call features is insufficient. However, in the post-mortem setting we can detect up to 58% of all fraud cases. At a false positive rate of 5%, we can detect up to 33% of all fraud cases in the pre-call setting, compared to 82% in the post-call setting. The true and false positive rates for the post-call features are comparable to earlier work by Kübler [38] and shows that, with respect to requirements **R1** and **R5**, Isolation Forests have potential for the detection of IRSF. Enabling detection of IRSF using only pre-call features, however, requires additional tuning in order to meet the requirements.

5.2 Experiment 2: Effect of parameter settings

This experiments is conducted to evaluate the effect of parameters `max_depth` and `sample_size` on the detection capabilities of Isolation Forests. To this end, we first computed the ROC curves for each value of `max_depth` while varying parameter `threshold` with steps of 0.1 and fixing the `sample_size` parameter to 256. We also

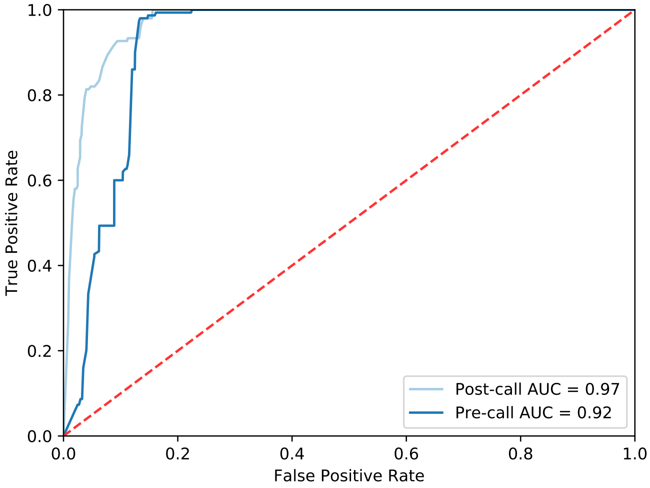

Fig. 6. Comparison of ROC curves for fraud detection using pre-call and post-call features. The dashed line represents the ROC curve for a random classifier.

computed the ROC curves for each value of `sample_size` while keeping the `max_depth` parameter fixed to $\log(sample\_size)$ and varying the `threshold` with steps of 0.1. For this experiment, we used pre-call features. Figure 7 reports the results of the experiment.

From the plot at the top of Figure 7, we can observe that, when `max_depth` is lower than 7, the detection rate drops significantly. We believe this is twofold. First, if the `max_depth` becomes smaller than the number of features, there is an increased risk of missing crucial feature splits during the tree generation phase of the algorithm. Second, in Section 4.1 we showed that anomalous data points occur alone in the leafs of an isolation tree. With a `max_depth` of 6, at most $2^6 = 64$ anomalous data points can be detected. Given that the dataset includes over 100 cases marked as `fraud`, the size of the tree is insufficient to assign every anomalous point to an empty leaf node. As a consequence, it is more likely for two inherently different data points to end up in the same leaf node, which leads to an overgeneralization of the resulting model.

A second observation is that increasing the `max_depth` does not significantly affect performance. By default, Liu et al. [25] recommend `max_depth` $= \log_2($`sample_size`$)$ to obtain a balanced tree. In our experiments this recommended value would correspond to $\log_2(256) = 8$. The idea of increasing the depth would be to give the algorithm more slack to add additional splits to better discriminate between normal and fraudulent points. Results show, however, that for `max_depth` $= 8$ the number of splits is already sufficient to identify the anomalies.

The plot at the bottom of Figure 7 shows that the increase of parameter `sample_size` does not lead to any significant improvement of the performance. This illustrates that taking a sample size of 256 is sufficiently large to obtain a data subset which is representative for the generation of one tree.

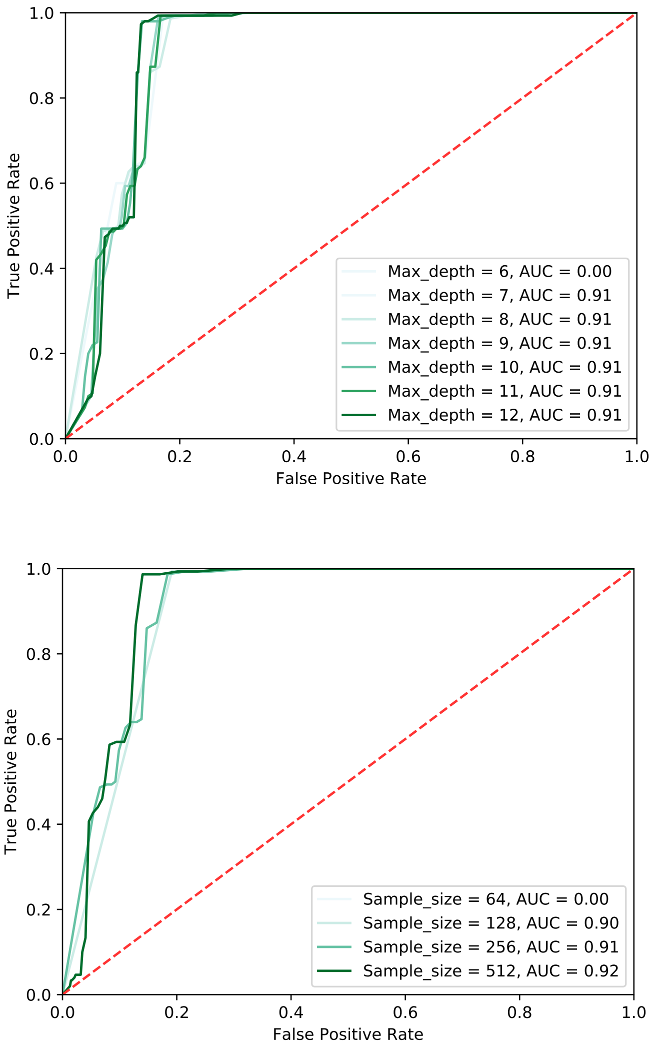

Fig. 7. Comparison of ROC curves for fraud detection using pre-call features at the variations of `max_depth` (top) and `sample_size` (bottom).

5.3 Experiment 3: Effect of different features

We conducted this experiment to investigate the impact of including certain features on the effectiveness of Isolation Forests for fraud detection. We compare different sets of features and measure the false positive and true positive rates. In particular, we consider the following feature sets:

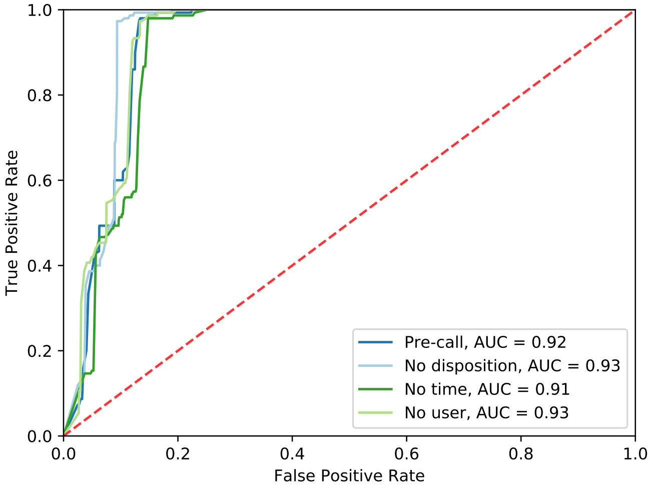

Fig. 8. Comparison of ROC curves for fraud detection using different feature sets. Specifically, the curve obtained using all pre-call features is compared to the curves obtained when either `disposition`, `time` or `user_id` are removed.

1. **Pre-call, no time**: `record_id`, `user_id`, `srcNr`, `srcCtry`, `dstNr`, `dstCtry`, `disposition`, which allows us to study the influence of the time of the call on the detection rate.
2. **Pre-call, no user**: `record_id`, `srcNr`, `srcCtry`, `dstNr`, `dstCtry`, `time`, `disposition`, which allows us to study the influence of the user identity on the detection rate.
3. **Pre-call, no disposition**: `record_id`, `user_id`, `srcNr`, `srcCtry`, `dstNr`, `dstCtry`, `time`, which allows us to study the effect of feature `disposition` on the detection rate.

We use the pre-call ROC curve from Experiment 1 as the baseline for the comparison. If a feature set performs as the pre-call feature set or better, then we conclude that the feature set is effective and the removed feature is not relevant for fraud detection; otherwise the feature set is not effective and the removed feature should be used for the classification of calls.

Figure 8 shows the result of the experiment. The removal of feature `disposition` has a slightly positive effect on the detection rate, as is the effect of removing `user_id`. A possible explanation for this could be that both features are not heavily correlated with fraud. Removing `time` has a negative effect on the detection rate, which coincides with the observation in Section 4.2 that the time of the call is a key factor for fraud detection. In the next experiment, we investigate how we can derive additional features from `time`.

5.4 Experiment 4: Effect of derived features

In this experiment we explore how to improve detection rates by deriving additional features from the call's start time field. We extract three kinds of features, namely:

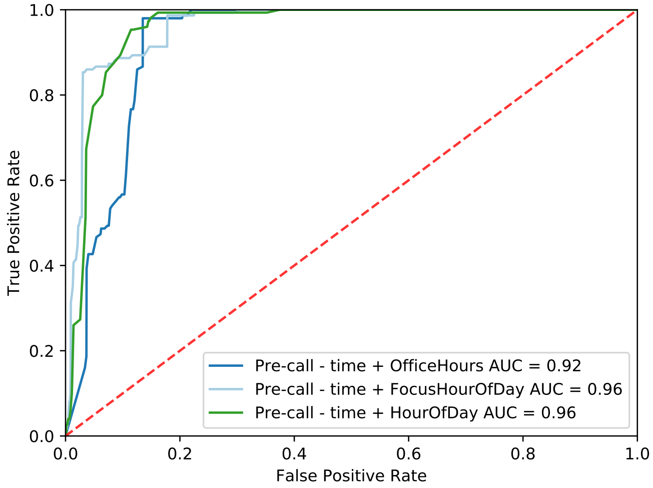

Fig. 9. Comparison of ROC curves for fraud detection using different discretization of feature `time`. Specifically, the curve obtained using all pre-call features is compared to the curves obtained when additionally to the pre-call features, features `OfficeHours`, `HourOfDay` or `FocusHourOfDay` are included.

1. **HourOfDay**: 24 Categorical features where every feature corresponds to a one hour interval (24 hours notation).
2. **OfficeHours**: A Boolean feature that indicates whether the call has happened between 09:00-17:00.
3. **FocusHourOfDay**: These features are similar to `HoursOfDay` but in this case only the hours just before 09:00 and just after 17:00 are considered, specifically, in the interval $[5, 9]$ and $[18, 22]$.

These features were introduced to exploit domain knowledge in the classification of calls. From the analysis of the dataset in Section 4.2 we observed that most fraud cases occurred outside office hours. We attempted to capture this information is several different ways. Initially, we only tried to capture this in a Boolean feature representing the office hour time-interval, i.e. `OfficeHours`. We stretched the idea of a Boolean feature per time interval all the way to including such a feature for every hour of a day. Finally, we focused this broad set of features to only including the time-intervals we know are correlated with fraud.

We applied Isolation Forests to the aforementioned feature spaces and compared the results to pre-call results in Figure 6. Specifically, in each experiment we consider the pre-call data set and replace the `time` feature with one of the derived features. The `time` feature is replaced to isolate the effect of including the derived features. The results are given in Figure 9.

We can observe that the features `HourOfDay` and `FocusHourOfDay` significantly improve the detection rates. In particular, with the latter we have a true positive rate of

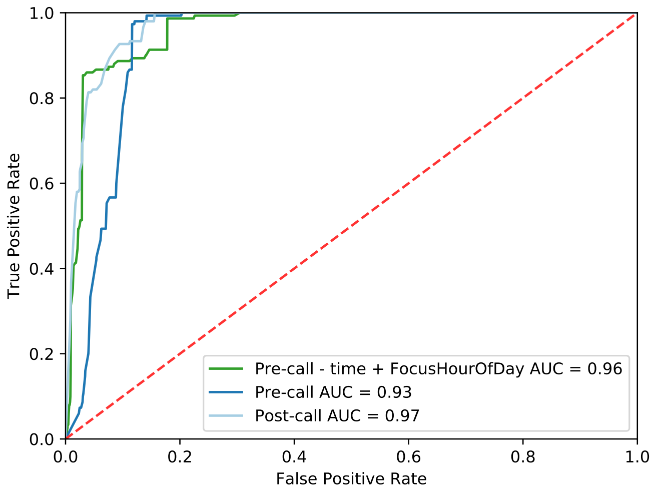

Fig. 10. Summary of the results. Specifically, the ROC curve for fraud detection using all pre-call features is compared to the ROC curve obtained using the feature `FocusHourOfDay`.

over 45%, at a false positive rate of 2%, which is comparable to the results obtained using post-call features. A possible explanation for this is that by increasing the number of features drastically, the chance of an isolation forest splitting on time increases. Since time is correlated with fraud, this could explain the performance increase.

6 Discussion and limitations

The experiments in Section 5 show that with careful choice of the parameters and feature space tuning, Isolation Forests provide a reliable tool for fraud detection. Figure 10 shows the false positive and true positive rates of Isolation Forests before and after the proposed optimizations as described in Experiments 1, 2, 3, and 4. Here, we can observe that at least 45% of the fraud can be detected reliably using the pre-call feature set at the cost of a 2% false positive rate and up to 87% at the cost of a 5% false positive rate. Given that the damage of such phone calls can be in the order of thousands of euros per call [15, 21, 30, 34], the discovery of a few fraudulent calls is already sufficient to cover operational expenses.

The introduction of time based derived features improves the detection and coincides with the observation that most of the fraud cases happened outside office hours. Experiment 2 has shown that decreasing `max_depth` does not yield a significant increase in performance. This is desirable since it gives less branching possible and clusters tend to become larger.

With respect to the requirements presented in Section 2, our evaluation shows that Isolation Forests are resilient to ill-balanced label distribution (**R1**) while enabling the support of categorical features with over 50 values per features (**R2**). Isolation Forests have a linear time complexity with a low constant and a low memory requirement which is ideal for high volume datasets (**R3**). Although the training of the forest was a matter of seconds, a larger data sample is needed to test whether requirement **R4** is fully satisfied. Experiments have also shown that, considering only pre-call features (**R6**), we are able to detect almost half of the fraud reliably (**R5**).

Isolation Forest do not limit itself to the analysis of features before call establishment, but can also be used for post-mortem analysis on CDR records. In Experiment 1 we showed that, for post-mortem detection, Isolation Forests shows promising results. However, more future work is required to explore the boundaries of the technique. The computation efficiency of the algorithm enables fast retraining of the model to avoid phenomena such as concept drift [36].

Although Isolation Forests have shown promising results in this application, the approach has some limitations. First, the construction of an isolation forest is done by randomly choosing features and splitting on values. Depending on the domain, however, certain splits might be more favorable than others. Currently, Isolation Forests do not allow enforcing certain splits or influencing the order in which features should be considered. One workaround would be to run the algorithm separately for every data subset of interest (e.g., per user); however, this no longer enables the discovery of any overlapping patterns between the subsets. Another workaround to enforce a data split would be to convert such split into a set of features (like it was done for feature `OfficeHours` in Experiment 4). This solves the problem only partially, since it is not possible to enforce the split at a certain location in the tree. Another limitation of Isolation Forests is that, like other unsupervised learning methods, this technique does not use labeled data in the training phase; therefore, it cannot leverage this information to learn how to distinguish between normal and fraudulent traffic.

7 Conclusions and Future Work

In this paper, we explored the boundaries of Isolation Forest for online detection of IRSF. We have performed a number of experiments to assess the effectiveness of the approach for both offline and online analysis of VoIP traffic and validated results against an existing industrial fraud detector. The results in Figure 10 show that Isolation Forest can identify up to 45% of IRSF traffic before the calls have been established. Since the proposed approach makes no underlying assumptions on the data to analyze, it is general and flexible enough to be applied in other domains such as uselogin analysis or clickstream analytics.

An interesting direction for future work is the study of other derived features, for example based on the countries involved. We also plan to perform a performance analysis of Isolation Forests on a larger-scale, e.g., with dozens of calls per second.

References

1. S. Ahmad and S. Purdy. Real-time anomaly detection for streaming analytics. *arXiv preprint arXiv:1607.02480*, 2016.
2. N. Alldrin, A. Smith, and D. Turnbull. Clustering with EM and K-means. Technical report, University of San Diego, 2003.

3. R. A. Becker, R. Cáceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. Clustering anonymized mobile call detail records to find usage groups. In *Workshop on Pervasive and Urban Applications*, 2011.
4. R. A. Becker, C. Volinsky, and A. R. Wilks. Fraud detection in telecommunications: History and lessons learned. *Technometrics*, 52(1):20–33, 2010.
5. P. Burge, J. Shawe-Taylor, et al. Detecting cellular fraud using adaptive prototypes. *AI Approaches to Fraud Detection and Risk Management*, pages 9–13, 1997.
6. A. Candel, V. Parmar, E. LeDell, and A. Arora. Deep learning with h2o. *H2O. ai Inc*, 2016.
7. B. Cappers. *Interactive visualization of event logs for cybersecurity*. PhD thesis, Technische Universiteit Eindhoven, 2018.
8. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.
9. Z. Ding and M. Fei. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17, 2013.
10. G. Dong and H. Liu. *Feature engineering for machine learning and data analytics*. CRC Press, 2018.
11. S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, 3(7), 2002.
12. S. B. Elagib, A.-H. A. Hashim, and R. Olanrewaju. CDR analysis using Big Data technology. In *International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering*, pages 467–471. IEEE, 2015.
13. S. Etalle. From intrusion detection to software design. In *European Symposium on Research in Computer Security*, pages 1–10. Springer, 2017.
14. P. Ferreira, R. Alves, O. Belo, and L. Cortesão. Establishing fraud detection patterns based on signatures. In *Industrial Conference on Data Mining*, pages 526–538. Springer, 2006.
15. FGSServices. Are you at risk from Toll Fraud? `https://fgsservices.co.uk/fgs-telecoms/systems/systems/toll-fraud/`, 2017.
16. M. Friedman. *There's no such thing as a free lunch*. Open Court LaSalle, IL, 1975.
17. C. Gibson. Europol Cyber Fraud Intelligence 2019 Report. `https://www.europol.europa.eu/sites/default/files/documents/cytel\_fraud\_intelligence\_notification.pdf`, 2019.
18. J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
19. I. Ighneiwa and H. Mohamed. Bypass fraud detection: Artificial intelligence approach. *arXiv preprint arXiv:1711.04627*, 2017.
20. P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Symposium on Theory of Computing*, pages 604–613. ACM, 1998.
21. Integrated Solutions. Evety Business needs to know Toll Fraud and VoIP. `http://www.integratedcom.net/every-business-needs-know-toll-fraud-voip/`, 2015.
22. Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang. Survey of fraud detection techniques. In *International Conference on Networking, Sensing and Control*, volume 2, pages 749–754. IEEE, 2004.
23. S. Kübler, M. Massoth, A. Wiens, and T. Wiens. Toll Fraud Detection in Voice over IP Networks Using Behavior Patterns on Unlabeled Data. In *International Conference on Networks*, pages 191–197, 2015.
24. M. Kumar, M. Hanumanthappa, and T. S. Kumar. Crime investigation and criminal network analysis using archive call detail records. In *International Conference on Advanced Computing*, pages 46–50. IEEE, 2017.
25. F. T. Liu, K. M. Ting, and Z. Zhou. Isolation forest. In *International Conference on Data Mining*, pages 413–422. IEEE, 2008.

26. N. Modani, K. Dey, R. Gupta, and S. Godbole. CDR Analysis Based Telco Churn Prediction and Customer Behavior Insights: A Case Study. In *International Conference on Web Information Systems Engineering*, pages 256–269. Springer, 2013.
27. D. Olszewski, J. Kacprzyk, and S. Zadrożny. Employing self-organizing map for fraud detection. In *International Conference on Artificial Intelligence and Soft Computing*, pages 150–161. Springer, 2013.
28. N. Pelroth. Phone hackers dial and redial to steal billions. `https://www.nytimes.com/2014/10/20/technology/dial-and-redial-phone-hackers-stealing-billions-.html`, 2014.
29. K. Peterson. *Business telecom systems: A guide to choosing the best technologies and services*. Crc Press, 2000.
30. S. Phithakkitnukoon, R. Dantu, and E.-A. Baatarjav. VoIP Security—Attacks and Solutions. *Information Security Journal: A Global Perspective*, 17(3):114–123, 2008.
31. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, et al. SIP: Session initiation protocol. RFC 3261, IETF, 2002.
32. M. Sahin, A. Francillon, P. Gupta, and M. Ahamad. Sok: Fraud in telephony networks. In *European Symposium on Security and Privacy*, pages 235–250. IEEE, 2017.
33. S. C. Tan, K. M. Ting, and T. F. Liu. Fast anomaly detection for streaming data. In *International Joint Conference on Artificial Intelligence*, pages 1511–1516. AAAI Press, 2011.
34. Tech Advance. Business At Risk of Toll Fraud. `https://techadvance.co.uk/blog/2018/05/businesses-at-risk-of-toll-fraud/`, 2018.
35. K. Wang, B. Wang, and L. Peng. CVAP: validation for cluster analyses. *Data Science Journal*, 8:88–93, 2009.
36. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.
37. A. Wiens, S. Kübler, T. Wiens, and M. Massoth. Improvement of User Profiling, Call Destination Profiling and Behavior Pattern Recognition Approaches for Telephony Toll Fraud Detection. *International Journal on Advances in Security*, 8(1 & 2), 2015.
38. A. Wiens, T. Wiens, and M. Massoth. A new unsupervised user profiling approach for detecting toll fraud in VoIP networks. In *Advanced International Conference on Telecommunications*, pages 63–69, 2014.