Privacy-Enhanced Fraud Detection with Bloom filters

Daniel Arp[1], Erwin Quiring[1], Tammo Krueger[2], Stanimir Dragiev[2], and Konrad Rieck[1]

[1] Technische Universität Braunschweig,
[2] Zalando Payments GmbH

Abstract. The online shopping sector is continuously growing, generating a turnover of billions of dollars each year. Unfortunately, this growth in popularity is not limited to regular customers: Organized crime targeting online shops has considerably evolved in the past years, causing significant financial losses to the merchants. As criminals often use similar strategies among different merchants, sharing information about fraud patterns could help mitigate the success of these malicious activities. In practice, however, the sharing of data is difficult, since shops are often competitors or have to follow strict privacy laws. In this paper, we propose a novel method for fraud detection that allows merchants to exchange information on recent fraud incidents without exposing customer data. To this end, our method pseudonymizes orders on the client-side before sending them to a central service for analysis. Although the service cannot access individual features of these orders, it is able to infer fraudulent patterns using machine learning techniques. We examine the capabilities of this approach and measure its impact on the overall detection performance on a dataset of more than 1.5 million orders from a large European online fashion retailer.

1 Introduction

The electronic commerce sector (e-commerce) is rapidly growing world-wide, offering a large variety of products which are delivered directly to the customers' home. In order to stay competitive with traditional shops, online retailers try to send out products as soon as possible after being purchased, thus leaving only little time to check for fraudulent activity. Following this strategy, the online merchant *Amazon* alone generated a sales revenue of about 177.87 billion dollars in 2017 [34]. However, the great success of these shops and their high incomes also attract cybercriminals that cause significant financial losses to the merchants.

The creativity of the cybercriminals is virtually unlimited and ranges from individual fraudsters refusing to pay for products to highly organized cybercriminals. So called *reshipping scams* are, for instance, a common fraud scheme which causes an estimated financial loss of 1.8 billion US dollars each year [14]. In these scams, the fraudsters use stolen payment data and let the shop send the products to middlemen who relabel the goods and forward them to the criminals.

In consequence, it becomes rather impossible for law enforcement to catch these cybercriminals due to the lack of any actual information about their identity.

As a reaction to the growing threat caused by cybercriminals, merchants have started to rely on fraud detection systems which automatically scan incoming orders for fraudulent patterns. According to a report published by LexisNexis [19], these systems often combine multiple fraud detection techniques, such as identity and address verification or device fingerprinting. Despite these efforts to automate the detection process, manual reviews are often additionally necessary to verify that an order is indeed malicious. Still, there remains a large number of undetected fraud incidents. As fraudsters tend to use similar fraud patterns among various merchants, an exchange of current fraud incidents between online retailers could effectively reduce the number of successful fraud attempts. In practice, however, this exchange of information is difficult because competitive merchants are often unwilling to share their data and also privacy laws pose a big hurdle for sharing customer data among different parties.

In this paper, we propose a novel approach that allows merchants to exchange information on recent fraud incidents without exposing customer data to other retailers. In particular, each merchant pseudonymizes incoming orders on the client side before uploading them to a central analysis service. This service in turn applies machine learning techniques to the pseudonymized data accumulated from all participating online retailers. In this way, the analysis service does not have access to orders in plaintext and each merchant cannot see data from the others. The resulting detection method, however, is capable of uncovering patterns in the pseudonymized data that may indicate global fraud and would have been missed otherwise.

Our pseudonymization method is based on Bloom filters as proposed by Schnell et al. [29]. We extend this data representation to improve the privacy of customers and empirically evaluate the probability of de-pseudonymization attacks. Based on these results, we calibrate the parameters of our pseudonymization method such that a machine learning algorithm can find actual fraud patterns while still providing a good protection of the underlying data.

We apply our method to a large data set consisting of more than 1.5 million actual orders collected by a large European online retailer and evaluate several learning methods on the pseudonymized data. We compare our results against a baseline that the merchant obtains without the use of pseudonymization. Although the detection performance decreases due to the information loss introduced by the pseudonymization, significant fraud patterns still remain in the data which can help to inform merchants about potential fraudulent activity.

In summary, we make the following contributions:

1. We present an approach that allows the sharing of data between different merchants without directly exposing sensitive information about their customers.
2. We determine the strength of the proposed pseudonymization method while assuming a realistic attack scenario.

3. We evaluate the detection performance of our approach on a large dataset containing 1,840,582 actual orders and demonstrate its ability to extract useful fraud patterns from the data despite the loss of information introduced by the pseudonymization.
4. To foster future research in this area, we make our method publicly available to the community[1].

The remainder of this paper is structured as follows. Section 2 provides some background information about the fraud ecosystem and common fraud patterns. In Section 3 we define a threat model which allows us to design a system for privacy-enhanced detection of online fraud. The resulting system is evaluated in Section 4. We discuss the challenges and limitations that we have faced throughout our research in Section 5 and discuss related work in Section 6. Section 7 concludes this paper.

2 Background

Online retailers are nowadays facing a large variety of different types of fraud. Due to convenience for the customers, it is not possible to simply enforce a strict verification process before delivering the purchase. Instead, the merchant needs to carefully weigh up the chance of losing a legitimate customer against the chance of being scammed by a cybercriminal. This decision is far from being trivial since fraudsters are continuously improving their patterns in order to remain undetected. In the following, we briefly discuss three prevalent fraud patterns of different complexity.

The so-called *chargeback fraud* [19, 38] represents a simple, yet common kind of fraud. A scammer purchases several products that are paid by credit card. After receiving the purchased goods, the fraudster requests a chargeback from her bank, thus getting the spent money refunded. This type of fraud understandably works just once at each merchant. Consequently, professional fraud often additionally involves identity theft where stolen credit card data or other personal information of other people are used to commit fraud repeatedly. Similar fraud activities also emerge in the context of bank transfers. For example, SEPA transfers can be canceled within a few days as part of a chargeback fraud.

Another type of fraud involves the payment by invoice, a popular payment method in some European countries. Normally, a customer purchases products that are delivered together with the invoice. This allows for *invoice fraud* which is similar to chargeback fraud in the sense that the payment is postponed to a later time. However, compared to chargeback fraud, it poses the additional risk to the retailer that no financial information about the customer is available—not even the minimal guarantee of a valid solvent bank account. This further lowers the threshold for committing fraud: while for chargeback fraud at least a (possibly stolen) credit card number is required, for the invoice the retailer has

[1] http://www.github.com/darp/abbo-tools

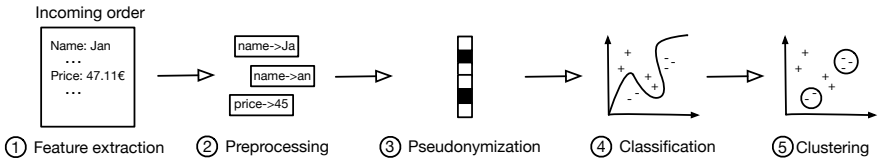


Fig. 1: Overview of the different steps of a privacy-enhanced fraud detection. A merchant (1) extracts features from an incoming order that are subsequently (2) preprocessed and (3) pseudonymized. Next, the analysis service applies (3) classification and (4) clustering methods to uncover fraudulent activities.

no interaction with the fraudster whatsoever. The fraudster obtains the products, but is never paying the invoice.

A more involved group of fraudulent activities combining various scamming patterns is known as *re-shipping scams* [14], commonly applied by professional cybercriminals. The fraudsters purchase goods from merchants by using stolen credit card data or benefit from deferred payment solutions like invoice. They hire middlemen commonly referred to as *drop points* via job announcements in newspapers or online portals. These drop points accept the packages and forward them to the fraudsters. The fraudsters' identity remains unknown while the possibly unwitting middleman might be approached by law enforcement. These middlemen are often used by multiple fraudsters to scam different merchants and are active for less than a month.

The exchange of fraudulent orders among multiple vendors and the application of a global classifier could effectively hinder fraudulent orders involving common drop points, stolen identities or credit card numbers. Overall, these fraud patterns highlight the need and benefit of a shared fraud detection that is discussed in the remainder of this manuscript.

3 Methodology

In this section we develop the overall setting of a shared analysis service among several merchants. We then derive a threat model and discuss the resulting privacy risks. Based on this step, we finally design a pseudonymization method to protect the customers' privacy during fraud detection.

3.1 The Analysis Service

Each participating merchant pseudonymizes its incoming orders before uploading them to the analysis service. A classification model trained on the pseudonymized data returns a prediction score which describes the potential risk of a submitted order. In contrast to a classifier solely trained on the data from a single retailer, the proposed classifier has access to the orders from all participating merchants.

In this way, it is capable to identify global fraud incidents that could be missed by a single vendor. As consequence of this design, the analysis service does not have access to data in plaintext and only the merchants can link reported fraud predictions to the original orders. That is, no information about ordered goods and customers are shared in clear with the analysis service.

Figure 1 summarizes the processing chain of the merchants and the analysis service. The features are extracted and preprocessed for each incoming order, pseudonymized and mapped to a vector space at the client side. Subsequently, the analysis service performs classification and clustering to identify fraud incidents. We discuss these steps in more detail in the following.

3.2 Features for Fraud Detection

To identify online fraud effectively, the classification model needs access to a set of discriminating features. The participation of a diverse range of online retailers also requires the definition of a meaningful subset of features that every retailer can contribute to. Thus, we focus on a minimal set of features which on the one hand are naturally available due to the purchase process and on the other hand enable the classification model to discriminate fraudsters from normal orders:

Address data. Every online fraud needs to be delivered to a certain physical address before the fraudster is able to resell the stolen goods and generate profit. The drop points are often reused since it is difficult to organize a multitude of delivery places without the help of a sophisticated organizational structure which is often not available. By collecting address data from multiple merchants it becomes easier to identify suspicious behavior for one particular address.

Cart items. The ultimate goal of the fraudster is to get goods for free which she can easily resell. In most cases she will therefore focus on specific brands and types of goods which have a good market value. This highly resaleable combination of goods in correlation with fraud will emerge naturally in the data pool of the analysis service and thus can be exploited by the classifier. We describe the ordered goods as a list of unique article identifiers and their respective prices.

Iterations. Fraudsters try to optimize their shopping cart by repeatedly adding or removing items until they can fool the checkout system and get the delivery. This is the single point where they can receive feedback from the fraud detection system and try to uncover the black box by exploiting common assumptions, for instance that a lower basket size increases the chance to get through.

Solvency score. Online retailers usually include a solvency score in their assessment of a customer. This score describes more or less accurately the probability that a customer will default. In the context of fraud detection, this feature helps to discriminate benign orders from fraud orders: If a customer has a good solvency score she is most of the time an actual person with a positive shopping history and is thus less likely to commit fraud.

This minimal set of features allows the analysis service to build classification models that balance out the amount of used features with the benefit of the pooling effect.

3.3 Threat Model

Sharing this kind of data between several parties obviously raises serious privacy and competition concerns. An order contains sensitive information about a customer such as her name, address and purchased products. To derive a secure pseudonymization method, we therefore need to define a threat model that describes the involved parties and their capabilities.

Merchants. The analysis service is used by multiple, possibly competitive merchants. A fraudulent merchant might thus try to abuse the analysis service to access confidential information from other merchants, such as the amount and type of commonly sold goods, the addresses of active customers and so on.

Hence, we design the analysis service such that the participating online retailers do not need to trust each other. In particular, each merchant has only access to its own uploaded data, that is, no retailer ever needs to have explicit access to the pseudonymized data of other participating merchants. Instead, the information of fraud incidents from other retailers is implicitly contained in the classification model trained by the analysis service.

Analysis service. A fraudulent operator of the analysis service has access to the data of all merchants, thus posing a serious risk to the confidentiality of the data. We assume that the operator is not one of the participating retailers but knows the names and addresses of some customers in the dataset. Using this information, she tries to deduce the goods that a particular customer has bought from one or several merchants.

In consequence, we have to ensure that the analysis service never has access to the plain data but only to pseudonymized orders. Still, the possibility is given that the operator of the analysis service attempts to break the pseudonymization using her background knowledge about certain customers. Thus, we need to strengthen our pseudonymization technique accordingly.

3.4 Pseudonymization

After discussing the utilized features and defining the threat model, we can finally develop a suitable pseudonymization technique. This technique has to fulfill certain requirements in our scenario. Most importantly, it should not be possible to easily reconstruct the information stored within the pseudonymized orders. At the same time, it should allow a machine learning algorithm to still extract fraudulent patterns from the data. Moreover, the approach should be capable of handling different data types as the discussion of the features in the previous subsection highlights.

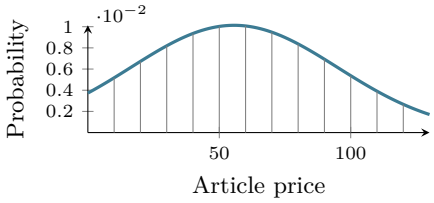


Fig. 2: Instead of using the exact value of the numerical features, their values are discretized by binning them.

Preprocessing the Data. Our proposed pseudonymization technique is based on Bloom filters [1], which we describe afterwards. The conversion of an incoming order into this data structure requires a preprocessing that can be divided into two distinct steps.

First, non-string values are converted into a string representation. For numerical features like the article price, this is done by simply binning their values. In particular, the size of these bins is selected regarding to the value distribution of the considered feature. Figure 2 depicts an example of this procedure for the *article price*. The selection of the bin size affects both the detection performance and the pseudonymization strength. By selecting a large bin size, more articles get assigned to the same price. This makes it harder for an attacker to derive whether the filter contains a particular article solely based on its price value.

In the second step, all strings are decomposed into smaller substrings before being inserted into the Bloom filter. Overall, different types of decompositions exist which can be applied.

- *Word Decomposition.* The order is split at the whitespaces and the resulting elements are inserted into the Bloom filter. While the decomposition of orders through this method is rather simple, it is not possible to match strings whose spelling only slightly differ.
- *N-Gram Decomposition.* In contrast to the word decomposition, the extraction of *n-grams* allows us to compensate for spelling mistakes and thus to decide whether two Bloom filters contain similar strings [8].
- *Entity Decomposition.* This decomposition is similar to the word decomposition, but additionally stores the information to which part of the order a particular word belongs. This, for instance, allows determining whether the shipping and billing address of an order differ—a pattern indicative for fraudulent activity.
- *Colored N-Grams Decomposition.* Similar to the decomposition in entities, colored n-grams store to which part of an order the extracted n-gram belongs. Figure 1 shows an example for a colored 2-gram decomposition.

Bloom filters. After the preprocessing of an order, the resulting strings are finally put into a Bloom filter. For each order, we initialize a separate Bloom

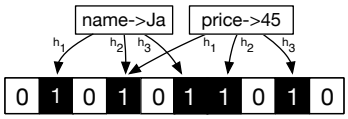


Fig. 3: Two elements are inserted into the Bloom filter using three hash functions (with a collision at the 4th bit).

filter. This probabilistic data structure enables storing large sets of elements within a limited amount of memory while simultaneously allowing an efficient comparison between different filters. At the same time, it does not allow an attacker to recover the information stored inside the data structure without background knowledge. Initially proposed for spell checking, Bloom Filters have already been successfully applied in several privacy-sensitive fields including the linkage of health records [11, 21, 26, 29].

Figure 3 depicts the basic concept behind Bloom filters schematically. The filter is a bit array of fixed length m where all bits are initialized as 0s. To insert an element x into the filter, a predefined number of k independent hash functions $h_i(x)$ are applied on the element. Each hash function maps the element to a particular position in the filter where the corresponding bit is set to 1. Similarly, it is possible to check whether the filter contains a particular element by applying these hash functions to the element and checking whether the corresponding bits are set to 1. If one of the bits is not set, the element has definitely not been inserted into the filter. In contrast, a positive match may be a false positive if the bits are set to 1 by other inserted elements.

These so-called collisions are usually an unwanted property of Bloom filters. However, collisions are desirable in our case since they already thwart an attacker from certainly reconstructing information stored within the filter. Nonetheless, this mechanism on its own is not sufficient to protect sensitive data as our evaluation in Section 4 underlines.

Hardening the Bloom filter. We examine several extensions of Bloom filters to strengthen their security properties.

- *Noise Insertion.* Adding noise to the Bloom filter can help to protect the data stored inside of it [11, 21] but can also partially destroy important information. We examine the effects of this approach for our application scenario by randomly setting bits in the filter.
- *Merging Filters.* Instead of just setting random bits, it is also possible to sample fake items from their respective distributions and add them to the Bloom filter. While this approach is more complex, it also further lowers the probability of successful frequency analysis attacks [16]. We implement a similar approach by merging multiple filters into a single one before sending it to the analysis service. Thus, an attacker has no possibility to assign a specific feature, e.g. an article, to a particular customer.

– *Keyed Hash Functions.* If an attacker has knowledge of the underlying distributions of the dataset and exact parameters used to pseudonymize the data, she can perform a dictionary attack and reconstruct the information stored inside the filters. This kind of attack can be effectively thwarted by keyed hash functions [28, 29]. In our case, the retailers share a secret key which is unknown to the operator of the analysis service, thus significantly improving the protection of the data stored inside the filters.

With these extensions at hand, it should be possible to clearly lower the probability of a de-pseudonymization. However, these techniques can simultaneously affect the detection performance of the classifier. We examine and discuss the effects of these techniques in Section 4.2.

3.5 Learning-based Fraud Detection

In the last step, we apply machine learning techniques for automatically detecting fraudulent patterns in the pseudonymized data. The usage of machine learning relieves a fraud analysis from manually constructing detection rules. In particular, we consider *classification* and *clustering* techniques. In the classification step, a learning model distinguishes between fraud and non-fraud cases. Afterwards, fraudulent patterns are extracted from the data by applying clustering techniques. This allows a fraud analyst to interpret these patterns and to take further actions if necessary.

Classification. The application of machine learning requires an appropriate vector representation of each Bloom filter. To this end, we associate each bit of the Bloom filter with a dimension in an m-dimensional vector space, where each dimension is either 0 or 1 and m corresponds to the length of the Bloom filter:

$$x \in \mathbb{R}^m = (b_1, b_2, \ldots, b_m)\,, \quad b_i = \{0, 1\}\,. \tag{1}$$

This yields very sparse high-dimensional data on which machine learning techniques can be applied. We examine the performance of *Linear Support Vector Machines* [10] and *Gradient Boosted Trees* [7] on this representation.

Clustering. In the next step, we try to find fraudulent patterns within the pseudonymized data by applying clustering methods such as k-means [9]. The identified clusters are ranked according to their ratio between fraud and benign samples. That is, clusters that contain many fraud incidents and preferably no benign samples are ranked at the top.

We can then extract (pseudonymized) fraudulent patterns from the highest ranked clusters. Figure 4 schematically visualizes an example for this process. Each Bloom filter of a fraud case is represented as a row in the left image. Red pixels represent set bits, black pixels unset bits. In addition, white horizontal lines separate the different clusters from each other. This representation easily uncovers fraudulent patterns as a unique combination of red vertical stripes in the image. In practice, the analysis service can extract these patterns and

send them to each online retailer. Since they have complete knowledge of the underlying pseudonymization technique, they are able to map back the fraud pattern to plaintext. Figure 4 shows an example on the right where an uncovered combination of n-grams indicates fraudulent activity.

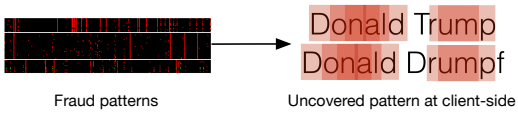


Fig. 4: Schematic visualization of the process to uncover fraud patterns on the client-side.

4 Evaluation

A successful operation of the central analysis service rests on two key requirements: First, we need to hinder a de-pseudonymization as good as possible. Second, we should be able to apply machine learning techniques to detect fraudulent orders. To evaluate whether we can balance these opposing requirements, we conduct the following experiments:

1. *Calibrating the data protection.* We examine the strength of our implemented pseudonymization method under the given threat model. Based on the results, we preselect a range of parameters that ensure a good data protection.
2. *Calibrating the detection performance.* We pseudonymize a sample of the data using the selected parameter ranges and train a classifier for each combination. We pick the parameter combination that yields the best detection results.
3. *Classification.* We pseudonymize the complete dataset of orders and evaluate the detection performance on this data. Subsequently, we compare the results with the detection performance achieved on the unprotected data.
4. *Clustering.* Finally, we cluster the pseudonymized data and extract common patterns of professional fraud from it. We then discuss how these patterns can help merchants to identify fraud more quickly.

4.1 Evaluation Dataset

Our dataset consists of 1,840,582 orders including 14,179 fraud incidents from 2016 provided by Zalando, a large European online fashion retailer. The data was carefully cleaned to ensure a high data quality. To discriminate between benign and fraudulent orders, we consider the actual payment. We flag each order as fraudulent that is not payed after three months. We have conducted our experiments in close consultation with Zalando. In each step, we have carefully followed German data privacy laws.

4.2 Calibrating the Data Protection

We first examine the pseudonymization method described in Section 3.4 to preselect a range of promising Bloom filter parameters that provide high pseudonymization strength.

Attack Scenario. To adequately evaluate the protection introduced by the proposed pseudonymization method, we consider the following attack scenario according to the threat model from Section 3.3. The analysis service represents the adversary and tries to reconstruct information stored within the Bloom filters.

Without background knowledge, such an attack is not possible. The adversary needs to know the parameters that have been used to create the Bloom filters, such as the type of hash functions. In addition, the service needs a list of possible addresses or articles. Without this information, a Bloom filter simply appears to the adversary as random bit sequence. Therefore, we grant the service full knowledge of the underlying method and assume that it has collected a list of customer addresses and possible articles, for example, by crawling the web. Its objective is now to gain knowledge about the shopping behavior of the customers in the dataset. In particular, the service wants to derive which goods a particular customer has bought.

With the necessary background knowledge the attacker can create own Bloom filters with the names and addresses of targeted customers and compare them with the pseudonymized orders. To this end, the adversary uses the *Jaccard similarity* [33] which is defined between two Bloom filters B_1 and B_2 as

$$J(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|} = \frac{|B_1 \wedge B_2|}{|B_1 \vee B_2|} \,. \tag{2}$$

$B_1 \wedge B_2$ represents the bitwise intersection, $B_1 \vee B_2$ the respective union between the two vectors. The attacker can now match two Bloom filters if their similarity score is greater than a particular threshold. After having identified a particular customer in one of the pseudonymized orders, the adversary can run a dictionary attack in order to determine which goods have been purchased by this customer.

Results. For measuring the influence of different pseudonymization parameters, we sample an artificial dataset consisting of 1,000 distinct orders. Using this data, we evaluate the impact of several Bloom filter parameters on the pseudonymization strength. The obtained results are averaged over 5 repetitions.

Decompositions. The results for different decompositions types are presented in Figure 5a. The plot depicts the fraction of correctly re-identified customers for different decomposition types depending on the Bloom filter length. For all examined decompositions, the attacker is able to re-identify the majority of customers even when a small Bloom filter size of 500 Bits is selected. Further reducing the size of the filters increases the collision probability and in turn also lowers the de-pseudonymization probability. However, the high number of collisions destroys valuable patterns for the detection of fraud at the same time.

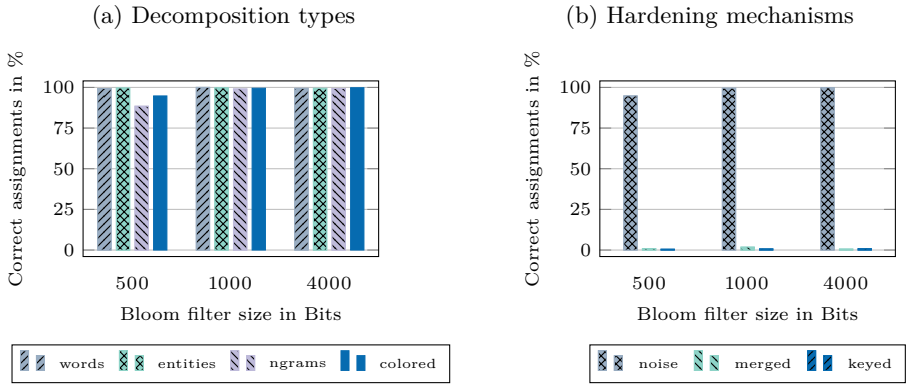


Fig. 5: Figure (a) and Figure (b) depict the impact of different decomposition types and hardening mechanisms on the pseudonymization strength.

Overall, we find that the collision probability does not provide proper protection of the sensitive data and we require further protection mechanisms. Moreover, the selected decomposition type has only little impact on the de-pseudonymization probability. Hence, we select two decomposition types that should allow deriving the best detection performance, i.e., *n-grams and colored n-grams.* While both allow handling spelling mistakes, colored n-grams also allow distinguishing between different parts of an order.

Hardening mechanisms. Since the collision probability does not provide sufficient protection, we have to rely on the hardening extensions described in Section 3.4. The results of their evaluation are depicted in Figure 5b. In this experiment, we add 10% of noise to the Bloom filters and measure the impact on the de-pseudonymization performance. Surprisingly, the addition of noise has almost no effect on the success of the attacker. The reason for this is that the attacker has knowledge about the name and address of a customer in our attack scenario. If both are re-identified in a particular Bloom filter, the probability is very high that the pseudonymized order indeed belongs to that customer—despite the presence of noise.

In contrast to adding noise, the two other hardening mechanisms succeed in protecting the customer data. If we merge k orders during the pseudonymization with $k = 3$, the attacker is unable to re-identify the order of a particular customer. However, if the merged order is identified as fraud, the merchant needs to check which one of the k orders actually contains fraud patterns. The hardening mechanism of keyed hash functions also successfully thwarts de-pseudonymization without the drawback of the merge method. In this case, the mechanism requires that the key remains unknown to the attacker.

In summary, the adversary in our attack scenario can be effectively thwarted when merging multiple orders or by using keyed hash functions. These two mechanisms provide a good protection independent from the size of the Bloom filters. In

the following, we thus examine the effects of both hardening mechanisms on the detection performance using Bloom filter sizes between 1000 and 10,000 Bits.

4.3 Calibrating the Detection Performance

The parameters selected in the previous step ensure a good data protection. It thus remains to calibrate our approach such that also a good detection of fraudulent activity is possible—if at all.

Overall, we have 384 different parameter combinations to evaluate after the preselection of parameters, such as the size of the Bloom filter, the regularisation parameter of the learning method and the lengths of the n-grams. In order to cope with this large number, we use only a small subset of the available training data and perform the model selection on it. This subset consists of 11,145 samples including 5,591 fraud incidents. We train a linear SVM on the data and measure its performance using the area under the ROC curve (AUC) [2]. We bound the AUC at 1% false positives to favor models with low false-positive rates. Having large false positive rates could otherwise lead to the rejection of legitimate customers, thus causing even greater financial loss to the merchants.

Based on the results of these experiments, we select a Bloom filter size of 4000 bits and a colored 2-gram decomposition. Moreover, we choose a bin size of 10 and 1 for the article price and the solvency score, respectively.

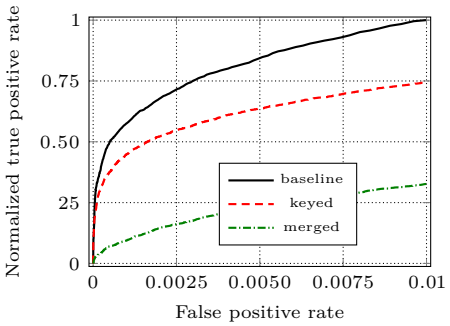
4.4 Classification

We finally examine the change in detection performance on the full dataset. We pseudonymize the dataset using the previously determined parameter values. We then split the dataset into two distinct sets and compare the detection performance obtained on the pseudonymized data with the original performance. The results are presented in Figure 6a. The baseline provided by Zalando is depicted in black whereas the results obtained on the pseudonymized data are shown as colored lines.

Note, that all classifiers have been trained on the same set of features in order to ensure comparability. Using keyed hash functions as hardening mechanism, we achieve a detection performance of about 75% compared to the results obtained by Zalando at 1% false positives. As can be seen from Figure 6a, this ratio remains nearly constant, even for significantly lower false positive rates such as 0.1%. We credit the difference in detection performance compared to Zalando to the information loss induced by the pseudonymization. While Zalando, for instance, trains the learning algorithm based on the exact numerical values, we lose information due to the binning of numerical features as described in Section 3.4. Nonetheless, we can uncover a large fraction of the fraud cases without access to the original orders, demonstrating that a central analysis service is technically feasible.

We also evaluate the detection performance after merging the filters. In particular, we randomly pick three Bloom filters and merge them into one. If at least one of the merged filters has been labeled as fraud, the resulting Bloom filter

(a) Classification results



(b) Clustering results

Rank	# Fraud	# Benign	Ratio
1	17	0	100.0%
2	31	0	100.0%
3	15	1	93.8%
4	357	41	89.7%
5	151	26	85.3%

Fig. 6: Classification results on 1,840,582 orders with a Bloom filter size of 4000 and two different hardening mechanisms. Each ROC curve shows the normalized true positive rate by using the performance of the baseline classifier as reference. Moreover, the table in Figure 6b presents the purity of the top-ranked clusters obtained on a dataset of 11,145 orders.

is also considered to be malicious. We notice a significant drop in the detection rate, thus achieving only about 30% of the original detection rate. We deduce that merging the Bloom filters changes the underlying distributions drastically and thus has a large impact on the detection rate.

In summary, we achieve a detection performance of about 75% compared to the unprotected data while at the same time clearly enhancing the protection of the underlying data. In the following, we evaluate whether it is possible to extract fraudulent patterns from the data using clustering despite the information loss introduced by the pseudonymization.

4.5 Clustering

We apply a k-means clustering to the dataset of 11,145 samples which has also been used to perform the parameter selection as discussed in Section 4.3. In particular, we test different values for k and pick the one which yields the best results, that is, a clustering where the top ranked clusters have the highest purity. Table 6b shows the top ranked clusters obtained when selecting $k = 100$.

We investigate these best-ranked clusters to determine whether they contain schemes of organized cybercrime. By de-pseudonymizing the data at the merchant, we find that the orders in the first three clusters are mainly grouped together due to specific articles or addresses they share. However, after consultation with Zalando, they can not be considered professional fraud and rather correspond to simple chargeback scams.

By contrast, cluster 4 and 5 exhibit typical patterns of professional scam. In particular, the forth cluster mainly contains orders of expensive clothes which

are delivered to drop points in Berlin. Moreover, these orders show a high iteration count, indicating that the fraudsters tried to optimize their shopping cart. Similar patterns can also be found in the cluster 5 where the fraudsters ordered rather high-priced accessories like watches or bags and let them send to drop points in Cologne. It is worth noting that both clusters contain 41 and 26 presumably legitimate orders, respectively. Overall, the case study thus shows that the extraction of fraud patterns from the pseudonymized data is possible, however, it requires tuning to lower the fraction of legitimate orders in large clusters.

5 Limitations

Our approach represents a first step towards a privacy-enhanced detection of fraudulent activity in e-commerce. However, there still exist several challenges and limitations which we discuss in the following, together with future research directions.

Malicious collaborations. In our threat model we do not consider the collaboration between a malicious merchant and a malicious analysis service. In this scenario, the key for the pseudonymization could be leaked to the analysis service, thus enabling its operator to run dictionary attacks on the Bloom filters. Fortunately, the collaboration of multiple merchants or a malicious analysis service alone do not pose a risk. It therefore remains future research to find extensions that also protect the customer data in scenarios where a malicious merchant and service collaborate.

Consistent data labeling. A consistent procedure for labeling the input data fed to the machine learning algorithm is essential to achieve a good classification performance. While this seems to be an obvious requirement, it is far from trivial in practice. This is because various online retailers often have their own definition of fraud and thus varying labeling procedures. In order to apply our approach in practice, it would be necessary that the participating online retailers agree on a common labeling scheme.

Data access. We only have access to the data of a single merchant to conduct our experiments. In order to demonstrate that our approach is indeed capable of identifying global fraud patterns, we thus require further data from other merchants. Still, the obtained results indicate that the identification of fraud is possible on pseudonymized data using our method and thus can help us acquire a larger group of participating merchants.

Frequency analysis attacks. Several researchers have shown that Bloom filters are prone to frequency analysis attacks [15–17]. Although these attacks pose a real threat in practice, they require the adversary to have exact background knowledge about the underlying distributions from which the features are drawn. While this is a realistic assumption for publicly available information such as names or addresses, it requires insider knowledge for other features like the solvency score.

By adding noise to the filters, the risk of a successful attack can be further reduced and should thus be negligible in our case. Nonetheless, measuring the actual risk needs further research since it highly depends on the particular application scenario and the knowledge available to the adversary.

6 Related Work

In the following, we discuss related work which contains research of mainly three different disciplines. First, we discuss research that provides insights into the underground ecosystem related to reshipping scams. Second, we describe papers that deal with fraud and malware detection. Finally, we review related literature which focusses on privacy-preserving technologies.

Underground Ecosystem. The first in-depth study on reshipping scams is presented by Hao et al. [14] who have analyzed the log files from seven reshipping scam operations that took place between 2010 and 2015. Their paper provides a detailed overview of the inner workings of this underground economy and estimates the overall financial loss caused by reshipping scams to be around 1.8 billion US dollars per year. In addition, they have been able to identify several possible ways how these criminal activities can be disrupted. However, the suggested countermeasures need to be enforced by the shipping service companies, thus requiring the online retailers to rely on these companies. In contrast, we focus on defenses that can be directly applied by the merchants themselves.

Other research groups have examined fraudulent activity closely related to reshipping scams. In particular, reshipping scams mostly imply identity theft [3, 30, 35] and mule recruitment [12, 22]. A survey on hijacking of online accounts for identity theft has been conducted by Shay et al. [30]. The authors have interrogated 294 people about their experience with account hijacking. Surprisingly, about 30.3% of the participants report that they have experienced compromise attempts on their email or social network accounts at least once. A similar study has been conducted by Bursztein et al. [3] but focusses on manual account hijacking. While identity theft allows fraudsters to distribute malware or spam using the stolen identities [18], it also poses a crucial part in reshipping scams. Consequently, some countermeasures initially proposed for spam or malware might also help to impede fraud in e-commerce.

Fraud detection in e-commerce. A large strain of research examines techniques to efficiently detect credit card fraud [5, 23]. Chan et al. [6] present a survey of different techniques for detecting credit card fraud. Likewise, other researchers have studied approaches to detect related fraud variants. In particular, Pandit et al. [25] propose a fraud detection system based on a *Markov Random Field* to discover fraud in online auctions. Their approach has been evaluated on a data set containing more than 60,000 actual users from eBay. Another method by Maranzato et al. [20] targets frauds against reputation systems in e-markets. An orthogonal strategy to defend against online fraud is the application of fingerprinting techniques like browser fingerprinting and device fingerprinting [4],

which unfortunately raises serious privacy concerns [24]. The most similar method to ours has been proposed by Preuveneers et al. [27]. The authors present a system which provides fraud detection as a service to the merchants. However, their approach does not consider data protection. Moreover, they use individual detection rules for each merchant instead of a global classifier trained on the data of several online retailers.

Privacy-preserving technologies. When processing personal data, it is particularly important to ensure that the data is protected from unauthorized access. Techniques to achieve a high protection level for sensitive patient data have been widely studied in the field of medical databases [16, 29, 37]. In particular, Schnell et al. [29] present an approach for privacy-preserving record linkage based on Bloom filters. Personal identifiers are stored in Bloom filters which can then be used to re-identify the database entry of a person within different databases without revealing its identity. Several researchers have demonstrated attacks on Bloom filters [15–17] using frequency analysis techniques. However, these attacks require the attacker to have background knowledge on the underlying distributions. While this is a realistic assumptions for publicly available information such as names or addresses, it requires insider knowledge for other features like the solvency score.

In addition, various researchers have recently demonstrated several successful information leakage attacks against machine learning models [13, 32, 36]. As a result of these attacks, the adversary is able to deduce some potentially sensitive information from the data that has been used to train the classifiers. In order to fend off some of these attacks, Shokri and Shmatikov [31] propose a system to jointly learn a neural network without exposing too much information of the local datasets. However, since random weights from locally trained neural networks are exchanged between the different parties, the exact privacy implications of this approach are still unclear. A similar defense technique has been presented by Wu et al. [39] to privately evaluate random forests and decision trees, but is limited to two parties and thus not applicable in our scenario.

7 Conclusion

This paper takes a first step towards an earlier detection of fraudulent orders committed against online retailers. As scammers often use similar strategies among several merchants, an exchange of information about recent fraud schemes between merchants could effectively impede the success of these scams. However, merchants are often unwilling to share this data with competitors and, moreover, have to follow strict privacy laws.

As a remedy, we propose an analysis service that allows multiple merchants to upload incoming orders that are pseudonymized in advance. In this way, the analysis service is able to extract global fraud patterns from the shared but pseudonymized data. This enables the service to inform the merchants about recent fraud schemes in a privacy-friendly way.

We implement a pseudonymization technique based on Bloom filters and evaluate its impact on the overall detection performance. To this end, we use a large dataset of actual orders collected by a large European online fashion retailer. In the pseudonymized setting we are able to spot 75% of the fraud cases detected by the privacy-unaware analysis at the same false positive rate. An additional clustering step further demonstrates that we are able to identify common patterns of professional fraud.

Although our approach does not provide perfect results, we demonstrate that balancing privacy and performance in fraud detection is technically feasible and direct access to sensitive information is not strictly necessary. Our approach is generic and can be extended using different pseudonymization techniques and learning methods. As a consequence, we are optimistic that future work can further narrow the gap between unprotected and privacy-enhanced fraud detection.

Acknowledgments The authors would like to thank Alwin Maier and Paul Schmidt for their assistance during the research project. Moreover, the authors gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) under the project ABBO (FKZ: 13N13634).

References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Communication of the ACM **13**(7), 422–426 (1970)
2. Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30**(7), 1145–1159 (1997)
3. Bursztein, E., Benko, B., Margolis, D., Pietraszek, T., Archer, A., Aquino, A., Pitsillidis, A., Savage, S.: Handcrafted fraud and extortion: Manual account hijacking in the wild. In: Proc. of Conference on Internet Measurement Conference (IMC) (2014)
4. Bursztein, E., Malyshev, A., Pietraszek, T., Thomas, K.: Picasso: Lightweight device class fingerprinting for web clients. In: Proc. of ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM) (2016)
5. Caldeira, E., Brandao, G., Pereira, A.C.M.: Fraud analysis and prevention in e-commerce transactions. In: Proc. of Latin American Web Congress (LA-WEB) (2014)
6. Chan, P.K., Fan, W., Prodromidis, A.L., Stolfo, S.J.: Distributed data mining in credit card fraud detection. IEEE Intelligent Systems **14**(6), 67–74 (1999)
7. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2016)
8. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science **267**(5199), 843–848 (1995)
9. Duda, R., P.E.Hart, D.G.Stork: Pattern classification. John Wiley & Sons, second edn. (2001)
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research (JMLR) **9**, 1871–1874 (2008)

11. Fanti, G., Pihur, V., Úlfar Erlingsson: Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. In: Proc. of Privacy Enhancing Technologies Symposium (PETS) (2016)
12. Florencio, D., Herley, C.: Phishing and money mules. In: Proc. of IEEE International Workshop on Information Forensics and Security (WIFS) (2010)
13. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In: Proc. of USENIX Security Symposium (2014)
14. Hao, S., Borgolte, K., Nikiforakis, N., Stringhini, G., Egele, M., Eubanks, M., Krebs, B., Vigna, G.: Drops for stuff: An analysis of reshipping mule scams. In: Proc. of ACM Conference on Computer and Communications Security (CCS) (2015)
15. Kroll, M., Steinmetzer, S.: Automated cryptanalysis of bloom filter encryptions of health records. In: Proc. of the International Conference on Health Informatics (HEALTHINF) (2015)
16. Kroll, M., Steinmetzer, S., Niedermeyer, F., Schnell, R.: Cryptanalysis of basic bloom filters used for privacy preserving record linkage. The Journal of Privacy and Confidentiality **6**(2), 59–79 (2014)
17. Kuzu, M., Kantarcioglu, M., Durham, E., Malin, B.: A constraint satisfaction cryptanalysis of bloom filters in private record linkage. In: Proc. of Privacy Enhancing Technologies Symposium (PETS) (2011)
18. Levchenko, K., Pitsillidis, A., Chachra, N., Enright, B., Félegyházi, M., Grier, C., Halvorson, T., Kanich, C., Kreibich, C., Liu, H., McCoy, D., Weaver, N., Paxson, V., Voelker, G.M., Savage, S.: Click trajectories: End-to-end analysis of the spam value chain. In: Proc. of IEEE Symposium on Security and Privacy (2011)
19. LexisNexis: True cost of fraud study (2016)
20. Maranzato, R., Pereira, A., do Lago, A.P., Neubert, M.: Fraud detection in reputation systems in e-markets using logistic regression. In: Proc. of ACM Symposium on Applied Computing (SAC) (2010)
21. Mor, N., Riva, O., Nath, S., Kubiatowicz, J.: Bloom cookies: Web search personalization without user tracking. In: Proc. of Network and Distributed System Security Symposium (NDSS) (2015)
22. Motoyama, M., McCoy, D., Levchenko, K., Savage, S., Voelker, G.M.: Dirty jobs: The role of freelance labor in web service abuse. In: Proc. of USENIX Security Symposium (2011)
23. Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y., Sun, X.: The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems **50**(3), 559–569 (2011)
24. Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., Vigna, G.: Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In: Proc. of IEEE Symposium on Security and Privacy (2013)
25. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: A fast and scalable system for fraud detection in online auction networks. In: Proc. of the International World Wide Web Conference (WWW) (2007)
26. Perl, H., Yassene, M., Brenner, M., Smith, M.: Fast confidential search for biomedical data using bloom filters and homomorphic cryptography. In: International Conference on eScience (2012)
27. Preuveneers, D., Goosens, B., Joosen, W.: Enhanced fraud detection as a service supporting merchant-specific runtime customization. In: Proc. of ACM Symposium on Applied Computing (SAC) (2017)
28. Schneier, B.: Applied Cryptography. John Wiley and Sons (1996)

29. Schnell, R., Bachteler, T., Reiher, J.: Privacy-preserving record linkage using bloom filters. In: BMC Medical Informatics and Decision Making (2009)
30. Shay, R., Ion, I., Reeder, R.W., Consolvo, S.: "my religious aunt asked why i was trying to sell her viagra": Experiences with account hijacking. In: Proc. of ACM Conference on Human Factors in Computing Systems (CHI) (2014)
31. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proc. of ACM Conference on Computer and Communications Security (CCS) (2015)
32. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: Proc. of IEEE Symposium on Security and Privacy (2017)
33. Sokal, R., Sneath, P.: Principles of Numerical Taxonomy. W.H. Freeman and Company (1963)
34. Statista: Net sales revenue of amazon from 2004 to 2017. https://www.statista.com/statistics/266282/annual-net-revenue-of-amazoncom/ (2018), last visited April 2018
35. Thomas, K., Iatskiv, D., Bursztein, E., Pietraszek, T., Grier, C., McCoy, D.: Dialing back abuse on phone verified accounts. In: Proc. of ACM Conference on Computer and Communications Security (CCS) (2014)
36. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction apis. In: Proc. of USENIX Security Symposium (2017)
37. Vatsalan, D., Christen, P., Verykios, V.S.: A taxonomy of privacy-preserving record linkage techniques. Information Systems **38**(6), 946–969 (2013)
38. Worldpay: Fragmentation of fraud (2014)
39. Wu, D.J., Feng, T., Naehrig, M., Lauter, K.E.: Privately evaluating decision trees and random forests. In: Proc. of Privacy Enhancing Technologies Symposium (PETS) (2016)