https://doi.org/10.1007/s10489-021-02812-6

Towards addressing unauthorized sharing of subscriptions

Wei Zhang[1] · Chris Challis[1]

Accepted: 1 September 2021 / Published online: 16 October 2021

Abstract
Subscription-based business is booming in recent years, especially in the entertainment sector such as video and music streaming. Usually one subscription account can be shared among family members for the convenience of subscribers. However, account sharing also creates challenges for service provider, as many account owners share their subscriptions outside of the household. The widely spread practice of unauthorized sharing causes huge revenue loss for service providers. However, service providers are very cautious to pursue violators because identifying unauthorized shared accounts is a challenging task. First, the sheer volume of unstructured and noisy data makes it prohibitive to manually process the data. Moreover, it is legitimate for family members to share an account from any location and use many devices as they want. It is tricky to differentiate between unauthorized and legitimate sharing. In this paper, we propose an efficient solution to address the account sharing problem. Based on usage log data, our solution builds user profiles by accumulating and representing geolocation and device usage information. Then we estimate the risk of unauthorized sharing by analyzing the usage pattern of each account. The proposed solution can identify a large number of shared accounts and help service providers to recoup a significant amount of lost revenue.

Keywords Account sharing · Unsupervised approach · User profiling · Geo-spoofing · User privacy

1 Introduction

Last decade has seen significant developments in the subscription-based business. For example, Microsoft launched Office 365 (now Microsoft 365) to transition its conventional sale of Microsoft Office to a subscription service. Adobe Photoshop and Acrobat have also become a mostly subscription-based business. The content streaming industry has seen even more growth, with big names such as Netflix, YouTube, Amazon Prime, and more recently, Disney+. The COVID-19 pandemic has further accelerated the growth of video streaming subscriptions. In order to attract more customers, service providers usually offer family sharing subscriptions. For instance, Microsoft offers the Microsoft 365 Family, Google has YouTube Premium and Netflix also has the Premium plan for family sharing. Note that these plans are intended for users within the same household. However, sharing is so easy in the internet era, many members are sharing their subscription accounts with users outside of their household. A poll by Thomson Reuters in 2014 found that 15% to 20% of millennials shared their accounts [1]. Another poll by Consumer Reports in 2015 [2] found that 46% of respondents who use a streaming service share their accounts with external users. More recently in 2018, a study by CNBC found that an estimated 35% of millennials share passwords for streaming services [16]. This unauthorized sharing of subscriptions results in a huge loss of potential sale for service providers. The loss could be over 100 millions of dollars per month for Netflix alone according to [15, 16]. In addition to owners willingly sharing their accounts, there are *fraudulent* sharing when accounts were hacked and sold. For example, after the launching of Disney+, thousands of accounts were up for sale on hacking forums [10]. Although this kind of fraudulent sharing happens less frequently, it often causes a single account being used by a large number of people, leading to excessive resource consumption. For the sake of simplicity, *account sharing* means exclusively unauthorized sharing (with external users) for

This article belongs to the Topical Collection: *Emerging topics in Applied Intelligence selected from IEA/AIE2020*
Guest Editors: Hamido Fujita, Philippe Fournier-Viger and Moonis Ali

✉ Wei Zhang
wzhang@adobe.com

[1] Adobe Inc, San Jose, CA, USA

the rest of this paper. Sharing within a household is noted as *legitimate sharing* to avoid confusion.

It is clear that service providers would benefit remarkably by restraining account sharing practices. However, they hesitate to address the problem due to multiple reasons. First, the sheer volume of data makes it hard to start: leading providers have millions of users and billions of connections each month. Second, the data is unstructured and noisy: user logs are plain text with lots of missing or inaccurate data. Last but perhaps the most tricky problem is the existence of legitimate account sharing at the same time. Family members are allowed to share one account from anywhere, either in home or another state, and use any device of their choice. The prohibitive cost of manual labeling and the high risk of falsely restraining legitimate accounts have long discouraged service provides from tackling the account sharing problem. Only very recently in 2021, it is reported [5, 9, 17] that Netflix is testing a crackdown on account sharing. The crackdown is carried out in a small scale. Netflix is very cautious about the practice, which demonstrated how sensitive and risky it could be. Nevertheless, they finally decided to tackle the issue. This reveals that the loss is too significant to bear, even for a company with a yearly revenue of 25 billion dollars.

In this paper, we propose a novel solution that utilizes customer log information for automatically identifying shared accounts. It handles both fraudulent sharing from hackers and intended sharing between friends. This will benefit service providers in two ways: (1) the opportunity of significantly growing their revenue: conversion of shared accounts to regular paid accounts will bring in millions of dollars because of their large customer bases; (2) restricting shared accounts that do not convert to paid accounts, as well as fraudulently shared accounts. This leads to server/network load reduction and significant cost cutting.

2 Existing work

Currently, the most popular way of restraining account sharing is limiting the number of concurrent sessions. In addition, some providers ask users to register a limited number of devices to their accounts. However, the first approach can adversely affect concurrent usage by family members, who are entitled to do so. In addition, limiting concurrent usage can be circumvented by sharing accounts at different time periods. Having to register a limited number of devices is not desirable either, as it will negatively impact customer experiences for multiple reasons. First, it is a hassle for the customers to do the registration. In addition, with the number of streaming-enabled devices available nowadays, it is easy to hit the limit. Finally, an account owner can sell/give the “registered” device to other people, so they can use the “registered” devices and easily defeat the rule.

There has been considerable research [4, 18, 19, 22] on modeling user behavior from session logs, for the purpose of improving recommendations. They mostly focus on identifying multiple users in an account, because recommendation systems will return inferior results when multiple users are mixed in one shared account. In [4], Bajaj and Shekhar proposed to use hierarchical clustering to combine similar channels into clusters, thus decomposing a single account into multiple personas so they can customize the recommendation for each person in the family. Verstrepen and Goethals [18] proposed a Dis-AMbiguating Item-Based (DAMIB) algorithm which originated from the item-based top-N collaborative filtering approach. It implicitly splits the shared account into subsets so that preferences from different users are not tangled together. The implicit split helps circumventing the task of estimating the number of users which is error prone. Based on the assumption that different users have distinct temporal usage patterns in Internet Protocol Television (IP-TV) services, Wang et al. [19] proposed to decompose an account into multiple virtual users. In the context of online flight recommendation, one account can also be used to book tickets for multiple persons (friends and family members, etc.). Zhao et al. [22] tackles the problem by using topic modeling for passenger prediction. There are a few papers [11, 20] which attempted to determine whether an account is shared by multiple users. Subspace clustering was used in [20] for identifying users in movie rating datasets. To find out if an account was shared by multiple users, they used an model selection approach based on Bayesian Information Criterion (BIC). Visual inspection was used to check the results due to the lack of ground truth. Jiang et al. [11] proposed to use affinity propagation [8] algorithm to discover the number of clusters when grouping sessions within an account, thus determining the number of users that shared the account. These existing approaches for identifying multiple users are not applicable for identifying unauthorized account sharing. Because they do not distinguish whether the multiple users are from the same household or not. The fact that one account is shared by multiple users does not mean that it is shared with external users. More often than not, they are shared within households.

A company named Synamedia unveiled a service called “Credentials Sharing Insight” at Consumer Electronics Show in 2019 [13]. Their solution is to cluster users based on their streaming behaviors using a number of factors, e.g., when and where an account is accessed, what content and what device, etc. It then looks for anomalies in user’s behaviors and determines the probability of sharing. They classify account sharing as two types: “casual” (sharing between friends and families) and “fraudulent” (sharing due to accounts being hacked). The "fraudulent" accounts will be targeted for restraining. Nevertheless, fraudulent sharing only takes a small portion of account sharing. Most account sharing happens between friends as many surveys have shown. It is unclear if their solution can differentiate account sharing among friends from sharing within households. There is no public available report about their performance. This is understandable as their data are private. In addition, without a significant amount of human-labeled data, performance evaluation is infeasible as there is no ground truth to compare against. However, without performance metrics or some other ways to demonstrate that the results are trustworthy, service providers would be hesitant to adopt a solution due to the risk of losing customers.

The crackdown [17] that Netflix introduced recently is a form of two-factor authentication. When a user logins into an account, he/she will be asked to verify with a code sent to the account owner. This could curtail the fraudulent sharing and discourage owner-allowed sharing to some extent. However, as long as owners don't mind passing that occasional code, the crackdown will be of no use. In addition, this adds some inconvenience to account owners as they also need to go through the verification process.

3 Our approach

All service providers have some logs about how their customers connect and use their service, because customers have to go through some authentication steps in order to use their service. In this paper, we will use the data from the TV Everywhere ecosystem (one major player in the video streaming business) as an example, although our solution can be easily adapted to other data. TV Everywhere is also known as authenticated streaming, where subscribers are authenticated and authorized to stream video from Multichannel Video Programming Distributors (MVPDs). Major MVPDs in USA all have millions of subscribers. For example, both $AT\&T$ and Comcast have 20+ million subscribers [7]. The TV Everywhere ecosystem has access to session logs of all subscribers. Each session log contains a slew of information such as account ID, location and some content information, as shown in Fig. 1.

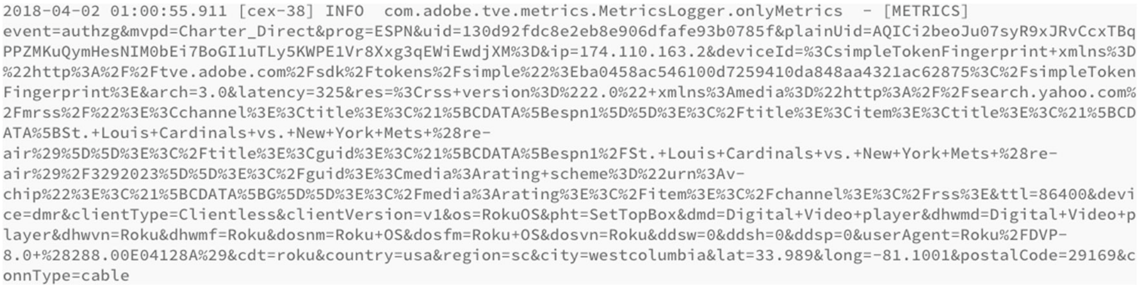
For anyone who tries to tackle the account sharing problem, a major challenge is that there is no ground truth label, since the labeling cost is prohibitive as we discussed in Section 1. This certainly limits our choice of algorithms. More importantly, a major issue arises: without any ground truth to compare against, how can we justify the results of the service? This is a critical question for demonstrating the value of our approach, due to the consequences of restricting account sharing: treating normal accounts as shared accounts (false alarm) will impact their customer experience and lead to potential loss of business. Classifying shared accounts as regular accounts, on the other hand, means the solution brings no value. We believe that only an **explainable** and **presentable** solution can address this challenge. Whether an account is identified as shared or not, it is paramount that service providers (e.g. MVPDs in the TV Everywhere ecosystem) can easily understand the reason so they can trust the results.

The account sharing detection solution must accommodate all variations that a normal account could have, so regular users are not impacted. After all, good user experience is the key for service providers to maintain and grow their customer bases. This means that it needs to handle the following scenarios and label them as normal accounts: 1. a big family with a large number of concurrent sessions, since everyone likes the freedom of choosing his/her own content; 2. ever-growing number of devices in a household as new devices are being added all the time; 3. family members commuting to places such as school/office/mall, or traveling to other states and accessing the service.

Although the problem is complicated, we have the following observations: First, non-family users (unauthorized sharing) are unlikely to share devices with account holders, since they live in physically different places. They might use devices that were owned by the account holder previously, through sale/gift, but the devices are transferred and not shared, so the account owner is unlikely to use them again. Second, non-family users typically access the service from different locations, rather than the home of account holders. Otherwise, they are more like a part of the household and virtually impossible to identify. Based on these principles, we propose to jointly utilizes geolocation and device information to estimate a sharing score for each account. Algorithm 3 describes how the sharing score is estimated. It depends on user profiles that are obtained from Algorithm 1 and Algorithm 2. Algorithm 1 scans through the log data, extracts user information and constructs efficient retrievable user profiles. Note the GPS coordinates in the session logs are usually noisy, so usage in one location, e.g. home, can have different coordinates (although ideally it should only be one). Algorithm 2 merges neighboring locations. Without this procedure, multiple geolocations might be associated with an account even if the owner only streams in her/his home.

3.1 Notations

Before presenting the algorithms, we first introduce the notation and related data structures, which are shown in Fig. 2.



```
2018-04-02 01:00:55.911 [cex-38] INFO  com.adobe.tve.metrics.MetricsLogger.onlyMetrics  - [METRICS]
event=authzg&mvpd=Charter_Direct&prog=ESPN&uid=130d92fdc8e2eb8e906dfafe93b0785f&plainUid=AQICi2beoJu07syR9xJRvCcxTBq
PPZMKuQymHesNIM0bEi7BoGI1uTLy5KWPE1Vr8Xxg3qEWiEwdjXM%3D&ip=174.110.163.2&deviceId=%3CsimpleTokenFingerprint+xmlns%3D
%22http%3A%2F%2Ftve.adobe.com%2Fsdk%2Ftokens%2Fsimple%22%3Eba0458ac546100d7259410da848aa4321ac62875%3C%2FsimpleToken
Fingerprint%3E&arch=3.0&latency=325&res=%3Crss+version%3D%222.0%22+xmlns%3Amedia%3D%22http%3A%2F%2Fsearch.yahoo.com%
2Fmrss%2F%22%3E%3Cchannel%3E%3Ctitle%3E%3C%21%5BCDATA%5Bespn1%5D%5D%3E%3C%2Ftitle%3E%3Citem%3E%3Ctitle%3E%3C%21%5BCD
ATA%5BSt.+Louis+Cardinals+vs.+New+York+Mets+%28re-
air%29%5D%5D%3E%3C%2Ftitle%3E%3Cguid%3E%3C%21%5BCDATA%5Bespn1%2FSt.+Louis+Cardinals+vs.+New+York+Mets+%28re-
air%29%2F3292023%5D%5D%3E%3C%2Fguid%3E%3Cmedia%3Arating+scheme%3D%22urn%3Av-
chip%22%3E%3C%21%5BCDATA%5BG%5D%5D%3E%3C%2Fmedia%3Arating%3E%3C%2Fitem%3E%3C%2Fchannel%3E%3C%2Frss%3E&ttl=86400&devi
ce=dmr&clientType=Clientless&clientVersion=v1&os=RokuOS&pht=SetTopBox&dmd=Digital+Video+player&dhwmd=Digital+Video+p
layer&dhwvn=Roku&dhwmf=Roku&dosnm=Roku+OS&dosfm=Roku+OS&dosvn=Roku&ddsw=0&ddsh=0&ddsp=0&userAgent=Roku%2FDVP-
8.0+%28288.00E04128A%29&cdt=roku&country=usa&region=sc&city=westcolumbia&lat=33.989&long=-81.1001&postalCode=29169&c
onnType=cable
```

Fig. 1 An example session log. Each session log contains comprehensive information including the subscriber's ID (obfuscated), the device used for connection, the GPS location, etc

1. *Device Usage Map (DUM)* represents a distribution of device usages. It is a hashmap $\mathcal{M} = \{(\mathcal{D}^0 : \mathcal{C}^0), (\mathcal{D}^1 : \mathcal{C}^1), \ldots, (\mathcal{D}^K : \mathcal{C}^K)\}$, K is the number of devices used in the location. $\mathcal{D}^k$ is the k^{th} device in the list, $\mathcal{C}^k$ is the count (histogram) of usage for the k^{th} device.
2. *Location Usage Map (LUM)* is a hashmap in the form $\{(\mathcal{L}_0 : \mathcal{M}_0), (\mathcal{L}_1 : \mathcal{M}_1), \ldots, (\mathcal{L}_N : \mathcal{M}_N)\}$, where N is the number of locations associated with the account. $\mathcal{L}_i$ is the i^{th} GPS coordinates, $\mathcal{M}_i$ is the *Device Usage Map* associated with $\mathcal{L}_i$. For instance, a LUM with only one element: $\{(40.2814, -111.6980) : \{(4277780 : 16), (11090085 : 2)\}\}$. This means that 2 devices have been used at location $(40.2814, -111.6980)$: device #4277780 was used 16 times (appears in 16 sessions), device #11090085 was used twice. Note we only keep up to 4 decimal places for GPS coordinates, i.e., locations are discretized. Because of this discretization, the minimal distance between any two locations is about 11 meters.
3. *userMap* stores the profile of all users; each entry contains a userID and a *Location Usage Map* associated with the account. It stores all locations and devices that are associated with the account, as well as the relationship between locations and devices (which devices are used in which locations). The relationship can be represented as a 2D (location-device) matrix, but the matrix will be very sparse: an account can have a long list of locations in its *LUM* due to traveling or account sharing; a location can have a long list of *DUMs* since a household can have an arbitrary number of devices. Therefore, we use a hashmap to represent these 3 levels of maps: *userMap*, *Location Usage Map* and *Device Usage Map*, so to make our approach very efficient (searching for their keys happens in a constant time).
4. *deviceMap* is a hashmap: { (original deviceID : device index), ... }. It represents a mapping from the the original deviceID (a string) to an integer value, e.g., 4277780 in the above example. Checking whether a

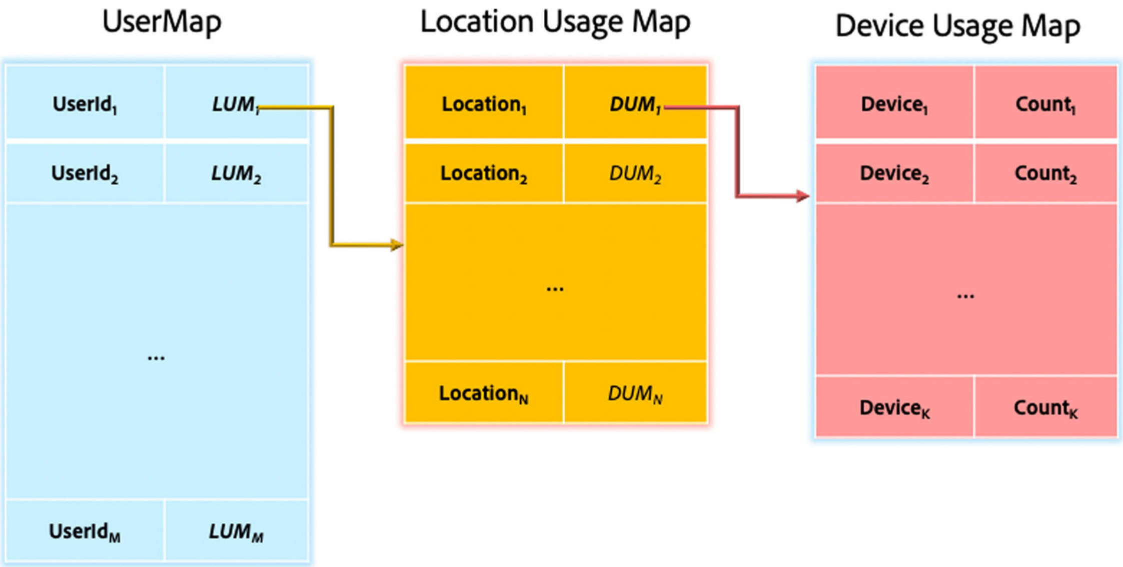


Fig. 2 The data structures used in the algorithms

device exists in the system is super efficient using hashmap. In addition, a device can appear in many locations, even across users. Using an integer instead of a string can reduce the space requirement significantly.

3.2 Algorithms

Our solution consists of four steps: 1, Scan log data and build initial user profile using Algorithm 1. 2. For each user, merge fragmented userMaps caused by GPS error based on Algorithm 2. 3. Estimate the sharing score for each user using Algorithm 3. 4. Visualize results.

Algorithm 1 Build initial user profiles (location and device graph).

```
1. Initialization: deviceMap ← {}, userMap ← {}, $nDevices \leftarrow 0$
2. Go through all session logs, one session at a time, to build userMap:
   (a) Extract usage information, e.g., deviceID and GPS coordinates $\mathcal{X}$.
   (b) if deviceID ∈ deviceMap then
           Get its device index $k$.
       else
           Add { deviceID : nDevices } to deviceMap
           $k \leftarrow nDevices, nDevices \leftarrow nDevices + 1$
       end if
   (c) if userID ∈ userMap then
           Merge to the existing entry in userMap:
           if $\mathcal{X} \in$ the associated LUM, say $\mathcal{X} = \mathcal{L}_j$ then
               if $k \in \mathcal{M}_j$ then
                   Increase usage count in the corresponding entry in $\mathcal{M}_j$
               else
                   Add a new entry $\{k : 1\}$ to $\mathcal{M}_j$
               end if
           else
               Add { $\mathcal{X} : \{k : 1\}$} to the LUM
           end if
       else
           Create an entry in userMap for this new account
       end if
```

Once we run through Algorithm 1, a userMap is built for each user. Each userMap has its *Location Usage Map*, associated with all the locations where the account has been used. Due to GPS noise, even if an account is only used at home, its *LUM* can have multiple entries, which in turn can adversely impact the sharing score estimation. Therefore, we use Algorithm 2 to address the noise in location.

Algorithm 2 Merge user profiles.

```
For every user in the userMap, do the following:
1. Multiply all entries in the LUM of the user with a smooth kernel.
2. Merge entries with the same coordinates.
3. Non-maximum suppression:
   Sort the result LUM by their number of device usages, in descending order.
   $i \leftarrow 0, P \leftarrow ||LUM||$
   while $i \neq P$ do
       for all $(\mathcal{L}_j : \mathcal{M}_j), j > i$ do
           if $dist(\mathcal{L}_i : \mathcal{L}_j) < \phi$ then
               Merge $\mathcal{M}_j$ into $\mathcal{M}_i$ (combine their device usage)
               Delete $(\mathcal{L}_j : \mathcal{M}_j)$
               P $\leftarrow P - 1$
           end if
       end for
       $i \leftarrow i + 1$
   end while
```

The kernel smoothing in Algorithm 2 step 1 spreads each entry of *LUM* to 5 entries. For example, for an entry of $((lat, lon) : \mathcal{M})$, after smoothing, it become 5 entries $\{((lat, lon) : \mathcal{M}_1), ((lat - 0.0001, lon) : \mathcal{M}_2), ((lat + 0.0001, lon) : \mathcal{M}_2), ((lat, lon - 0.0001) : \mathcal{M}_2), ((lat, lon + 0.0001) : \mathcal{M}_2)\}$). $\mathcal{M}_1$ is similar to $\mathcal{M}$, except that for each entry in $\mathcal{M}_1$, the count is half of the corresponding entry in $\mathcal{M}$. As mentioned before, lat and lon have 4 decimal places. By adding/subtracting 0.0001 in GPS coordinates, the four new entries are located right next to the original coordinates (lat, lon). For their *Device Usage Map*, the value of each entry in $\mathcal{M}_2$ is $1/8$ of the corresponding entry in $\mathcal{M}$. So the total usage count of these 5 entries equals to the total count in the original entry $((lat, lon) : \mathcal{M})$, only they are distributed among these entries.

After smoothing, there will be many entries with the same coordinates (because neighboring entries spread to each other). Then we merge them so each location only has one *Device Usage Map*, followed by non-maximum suppression. These steps help to handle fluctuations in GPS coordinates, by combining fragmented entries (due to error in coordinates) in the Location Usage Map into one single entry. Consider an over-simplified example: for a certain user who only accesses service from home, 3 entries are included in the associated *LUM* because of GPS noise, including one in the true location, one on the left and one on the right. After running through Algorithm 2, there will be only one entry remaining, with the true location. ϕ is set to be 50 meters in our experiment. The average GPS accuracy is about 7.8 meters [3], well within the ϕ range. This ϕ setting is also fine enough to identify accounts that are shared with non-family members, as it is unlikely that non-family members live within 50 meters.

Assuming the GPS noise follows a Gaussian distribution with zero mean $\mathcal{N}(0, \sigma)$, the observed coordinates will be scattered around the true coordinates. Algorithm 2 will merge them to a single entry, whose coordinates are likely the true value. If the GPS noise follows a non zero-mean Gaussian distribution $\mathcal{N}(\mu, \sigma)$, meaning there is a systematic error associated with the GPS, all coordinates will be shifted by μ. Nevertheless, the coordinates in the log of the same household will be scattered (due to noise) around that shifted center location. The drift will not affect the scoring algorithm since Algorithm 3 is only based on the usage pattern and *relative* distances, not the absolute location.

Algorithm 3 Sharing score estimation for each user in userMap.

```
1. Given $[(\mathcal{L}_0 : \mathcal{M}_0), (\mathcal{L}_1 : \mathcal{M}_1), \ldots, (\mathcal{L}_N : \mathcal{M}_N)]$, Set the base location $\mathcal{L} \leftarrow \mathcal{L}_0$.
2. Initiate a "registered" device map $\mathcal{M} \leftarrow \mathcal{M}_0$. Initiate risk score $R \leftarrow 1$.
3. For $i = 1, 2, \ldots, N$:
   (a) Calculate distance weight: $W_i^d \leftarrow log_\alpha(max(dist, \alpha))$, where $dist \leftarrow |\mathcal{L}_i - \mathcal{L}|$.
   (b) Let $\mathcal{M}_i$ be $\{(\mathcal{D}_i^0 : \mathcal{C}_i^0), (\mathcal{D}_i^1 : \mathcal{C}_i^1), \ldots, (\mathcal{D}_i^J : \mathcal{C}_i^J)\}$, where $J$ is the number of devices associated with the $i^{th}$ location. $\mathcal{D}_i^j$ is the $j^{th}$ device ID, $\mathcal{C}_i^j$ is the count of usages for the $j^{th}$ device.
   (c) $R_i \leftarrow 0$
       for $j = 0, 1, \ldots, J$ do
           if $\mathcal{D}_i^j \in \mathcal{M}$, say $\mathcal{D}_i^j = \mathcal{D}^k$ then
               $r \leftarrow \frac{\mathcal{C}_i^j}{\mathcal{C}^k}$
               $R_i \leftarrow R_i + \frac{1}{e^{-(r-\beta)/3}+1}$
               $\mathcal{C}^k \leftarrow \mathcal{C}^k + \mathcal{C}_i^j$
           else
               $R_i \leftarrow R_i + 1$
               Add $(\mathcal{D}_i^j, \mathcal{C}_i^j)$ to $\mathcal{M}$
           end if
       end for
   (d) $R \leftarrow R + R_i * W_i^d$
4. $W^t \leftarrow log_\gamma(max(t, \gamma))$, $t$ is the number of devices used by the account.
5. $R \leftarrow R * W^t$
6. Sharing score $S \leftarrow 2 * \frac{1}{e^{-(R-1)}+1} - 1$
```

The userMap generated from Algorithm 1 captures the usage pattern of all accounts: one entry for an account. Each entry contains a userId and its Location Usage Map, which includes a list of locations ordered by their device usage (from large to small). We use the Google Map API for visualizing the usage pattern, so it can be easily seen why an account is labeled as normal or abusive sharing. Some screen captures of the interactive map are shown in the results section, such as Table 1, Tables 7 and 9. Each location associated with the account is tagged with a red balloon. A red circle is centered at the root of each balloon, representing the usage in that location. The bigger the circle, the larger the number of uses (sum of all device usages in that locations). Note the circle size is not linearly proportional to the number of uses because the range can be very wide, from 1 to several thousands. Instead, the size is based on the natural logarithm of the numerical value, so that we can see the difference in usages across different locations. The location with the most uses is called the *base location* of the account. The base location is important for the scoring Algorithm 3. A regular account is more likely to have a dominant base location, because that is the place where most household members access the service. For a heavily shared account, the usage pattern is more distributed.

3.3 More on the scoring algorithm

Algorithm 3 estimates the score (risk) of an account being shared, by checking the device usage of all locations other than the base. If a device is not in the list of "registered" devices, i.e., it is a new device never used in the base location, it is more likely to be used by outsiders (users not belonging to the household). If it appears in the list, but used much more often in other locations than the base, it is also possibly an outsider's device (e.g. a friend who visits occasionally), although the probability is much lower than an "unregistered" device. This is captured in step (4) of Algorithm 3. We believe that the risk of sharing is fairly low when the non-base usage is not significantly higher than the usage at the base location. β is set to be 20 in our experiments. That is, when the usage in other locations is 20 times as high, R_i is increased by 0.5. R_i represents the sharing evidence from devices associated with the i^{th} location. Higher R_i leads to higher R, which ultimately leads to a higher sharing score. $r - \beta$ is divided by 3 so the logistic curve does not saturate too quickly. We apply the logistic function when updating R_i, so that the influence from each device usage is bounded. In other words, an extreme usage of one device at one location will not change R_i too much. High sharing score comes from consistent high usage of multiple "unregistered" devices. We are conservative in triggering higher sharing scores so

Table 1 This account has been used by 27 devices in total. They are all used in the base location and the number of devices is not super high, so it is still likely a case of family sharing. The sharing score is not 0, but relatively low at 0.05

Usage visualization	Location usage map
	{(39.9194, -75.4205): {11687811: 2, 2217860: 7, 12263382: 5, 1016583: 57, 77: 10, 578254: 3, 17937: 20, 10821077: 4, 3304278: 3, 11051844: 2, 12391413: 4, 10117597: 2, 8262432: 3, 2583585: 22, 6699239: 1, 6699240: 5, 1317930: 25, 2012653: 35, 3827630: 1, 11688048: 4, 3908979: 1, 12391412: 1, 1015669: 11, 5412342: 12, 2622200: 16, 12721209: 2, 10457717: 2}}

MASSACHUSE
Pro
CONNECTICUT
PENNSYLVANIA
rgh
New York
Philadelphia
MARYLAND
NEW JERSEY
Washington
DELAWARE

that the solution is not sensitive to isolated numerical errors in the log. Nevertheless, we are still able to identify millions of shared accounts as shown in Section 4.

We also take the distribution of locations into account when estimating sharing scores. The idea is that uses far apart are more likely to be due to account sharing. Household members may go to work or school and stream video every day, but are less likely to go to the other side of the country. The distance weight W^d is introduced for this purpose. The minimum distance α is set to be 50 so the distance weight will have no effect in Algorithm 3 (3d) for usage within 50 miles, while usage in locations which are hundreds of miles away will be penalized and lead to higher scores.

Even if all user sessions appear to happen in the base location, it is still possible that the account is shared since people can fake their location by geo-spoofing. For example, geo-spoofing has been used by *Pokemon Go* players to "go" to places without physically being there. It is not a widespread practice yet, but we should be ready to tackle it. The device weight W^t in Algorithm 3 is designed for this purpose. The higher the number of devices, the higher the weight. So even if all streaming sessions share the same location, the score will still be higher if there is an extremely large number of devices. This is our first attempt to address the geo-spoofing problem, so we are relatively generous on the parameter setting, with $\gamma = 20$. For example, based on the current settings, if 400 devices are used in an account, the weight will equal to 2. If the account's *Location Usage Map* has only one location (all sessions happen in one location), the final score will be 0.46. If the number of devices is less than γ, the weight is always 1. For accounts with just one location and less than 20 devices, their R value in Algorithm 3 (5) is always 1. Consequently, they get score 0, which means an unquestionably safe account. Even with this generous setting, we identified some potentially shared accounts where all sessions appeared to be in one location; an example is shown in Table 2. The worst case has 6,829 devices under one account, clearly a shared account using geo-spoofing. It gets a score of 0.75, not extreme but high enough to be identified.

When calculating the distance and device weight, we use logarithm instead of the original value. This represents the empirical knowledge that larger values indicate higher risk of sharing but the risk doesn't grow linearly. For example, distance changing from 50 to 100 miles is quite significant. Because 50 miles is likely within the range of local commuting (e.g., family members go to work), while 100 miles is probably not. On the other hand, distance changing from 1000 to 2000 miles is much less significant in terms of risk. They are both far away and require air travel, thus doubling the distance only leads to minor increase in sharing risk. When estimating sharing score, we use the logistic function so that it is bounded between 0 and 1.

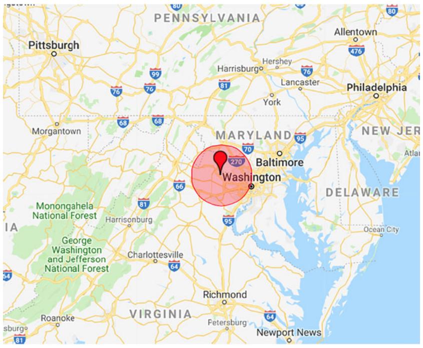
4 Results and discussions

We use a three-month session log of the TV everywhere system to illustrate the proposed solution. The number of users in this data is 30,620,878. The total number of session records in the data is 1,032,254,858 and the size of this data is 1.01 terabytes. Using 0.5 as the threshold for sharing score, we identified about 6.45% of accounts as shared, with very high confidence. It is usually straightforward to see that they are shared, as we show in Tables 7 and 9. Using a much lower (loose) threshold of 0.05, we identified about 15.66% accounts as shared. Some manual verification might be needed for the these results. Nevertheless, based on some random checks, we can see that most of them are indeed likely shared accounts.

4.1 Accounts with only one location

About 70% of users stream videos from just one location using less than 10 devices. They are all labeled as regular/non-sharing accounts as their sharing scores are 0. The accounts which have zero sharing scores all have the same visualization: one location with a few devices. Note that having only one location associated with one account does not necessarily mean the account usage is legitimate.

Table 2 All sessions happens in just one location for this account. However, the account has been used by many more devices, 72 in total. The number of devices suggests that geo-spoofing is probably used. The score is relatively high at 0.21

Usage visualization	Location usage map
 PENNSYLVANIA Pittsburgh Allentown Harrisburg Hershey Lancaster Philadelphia York Morgantown MARYLAND NEW JER Baltimore Washington DELAWARE Monongahela National Forest Harrisonburg Ocean City George Washington and Jefferson National Forest Charlottesville Richmond Roanoke VIRGINIA Petersburg Newport News	{(39.0329, -77.4866): {6435225: 4, 7796609: 2, 13210498: 2, 5182979: 4, 8522116: 2, 8911041: 3, 11916204: 2, 12181898: 1, 13113230: 2, 12885648: 2, 6745537: 2, 9590035: 1, 9984404: 2, 6644503: 1, 7834904: 2, 9573145: 2, 8482497: 2, 2710447: 2, 9993242: 2, 7778113: 2, 7055009: 3, 13225123: 2, 8222628: 1, 6520613: 2, 7564072: 3, 3754666: 2, 7653420: 2, 11134208: 2, 8911023: 2, 2710448: 2, 12746418: 2, 12634526: 2, 7379895: 4, ... } }

There are many cases with a significant number of devices, although each of them only has one associated location. Two examples are shown in Tables 1 and 2: both have only one location for all sessions. However, geo-spoofing might be used in these two cases. The case in Table 2 is more likely to be account sharing, because an extraordinary number of devices is used.

4.2 Accounts with multiple locations

Having a large number of locations associated with one account does not necessarily mean the account is shared. It can be caused by family members traveling.

4.2.1 Low risk accounts

The cases in Tables 3, 4 and 5 represent an interesting usage pattern: many locations and few devices. We call them *traveling accounts* and they have very low sharing scores. For the case in Table 3, the account has two devices (#747409 and #868586) associated with it. The base location is (40.7046, -73.9216) where device #747409 was used 15 times and device #868586 was used 6 times. Both devices were used in other locations, suggesting that they were taken to travel around. The account sharing score is very low for this case (only 0.01) and it is labeled as a safe account. The score is not 0 though, because it is possible that a friend visited the account holder's base location (probably home) multiple times with device #868586, thus he/she got the device "registered" to the account and lowered the account sharing score. Nevertheless, the probability is very low in comparison with other shared accounts. Note the sharing score is updated monthly with incoming data, so next month device #868586 will not be "registered" with the account if the friend no longer brings it to the account holder's base location. As a result, the sharing score would be much higher according to Algorithm 3, as the device is not used in the base location. This would cause the account to be labeled as shared, which is the desired result. Therefore, even if users know about how we identify shared accounts, it is not easy for them to game the system: they have to pay regular visits to the account holders' base location, in order to "register" their devices to the account. Otherwise, they will be identified.

The case in Table 6 represents another interesting type of usage pattern. Only 3 devices were used for accessing this account. However, the usage of the device #1858906 was higher in other locations than in the base location. So it might be a friend visited the owner's home with the device (unauthorized sharing). But it is equally likely that this is legitimate as the owner could use the device more at other places. Notice the distances between different locations are rather large, this slightly increases the likelihood of sharing. Overall, this is borderline case with a sharing score of 0.11. It needs further human verification (probably keep monitoring for a longer period of time for better decision).

4.2.2 Shared accounts

We did find millions of accounts which are almost certainly shared. See Table 7, Tables 8 and 9 for some typical cases. Both accounts in Tables 8 and 9 have sharing scores of 1, the highest possible score, meaning they are definitely being shared outside of family. The account in Table 7, although not as wildly shared as the other two cases, is a shared account as well: it is virtually impossible for a family to have such a usage pattern.

4.3 Discussion

As we have argued, the solution has to be **explainable** and **presentable** so people can understand and trust it. This

Table 3 This account has been used in many locations but is identified as a regular account. Because few devices were used and there is one dominating base location. It is likely a family member traveling with a mobile device. The base location with most usage is labeled bold

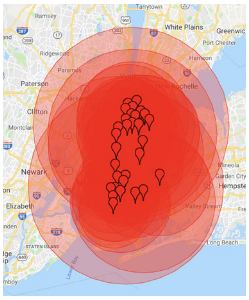
Usage visualization	Location usage map
	{(40.6797, -73.9503): {747409: 2}, (40.859, -73.8908): {747409: 1}, (40.8371, -73.8807): {868586: 1}, (40.679, -73.9618): {747409: 1}, (40.809, -73.9168): {868586: 1}, (40.8276, -73.896): {747409: 1}, **(40.7046, -73.9216): {747409: 15, 868586: 6**}, (40.8187, -73.8572): {747409: 3}, (40.728, -73.9493): {747409: 1}, (40.6936, -73.9265): {747409: 1}, (40.6471, -73.9549): {747409: 1}, (40.7903, -73.9468): {747409: 7}, (40.8202, -73.9202): {747409: 3}, (40.8464, -73.9027): {747409: 4}, (40.7758, -73.8749): {747409: 5}, (40.7608, -73.9457): {747409: 1}, ... }

Table 4 This account was accessed from many locations across a large region. The location with most usage is labeled bold. The account is identified as a regular account: only 2 devices were used and the base location is evident. It is likely due to a family member traveling

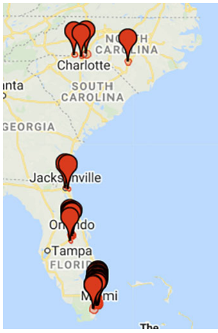
Usage visualization	Location usage map
	{ **(25.6611, -80.4076): {3279424: 2, 554707: 15}** , (26.0168, -80.1511): {3279424: 1}, (25.9929, -80.2742): {554707: 3}, (26.249, -80.2069): {554707: 1}, (26.1103, -80.27): {554707: 1}, (30.3053, -81.5117): {554707: 1}, (26.1405, -80.1738): {554707: 3}, (25.7003, -80.4084): {554707: 1}, (25.8534, -80.237): {554707: 1}, (26.0882, -80.1845): {554707: 1}, (25.6611, -80.4076): {554707: 7}, (25.8901, -80.3415): {554707: 3}, (26.1818, -80.2276): {554707: 4}, (25.7669, -80.2387): {554707: 5}, (25.8753, -80.2018): {554707: 1}, ... }

Table 5 The account has been accessed from a large number of locations. There is no clear base (heavily used) location: usage from all locations are similar. However, there is only 1 device ever being used. So it is the owner traveling

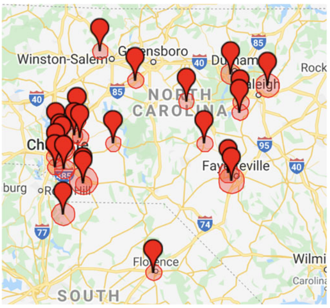
Usage visualization	Location usage map
	{(34.7094, -80.7751): {361260: 3}, (35.2285, -80.8449): {361260: 2}, (35.3438, -80.2263): {361260: 1}, (35.1177, -80.7602): {361260: 2}, (34.9837, -80.5492): {361260: 1}, **(35.011, -80.5512): {361260: 5**}, (35.0818, -78.9624): {361260: 2}, (35.414, -80.8526): {361260: 1}, (35.1331, -80.8597): {361260: 1}, (35.9646, -78.9405): {361260: 2}, (35.2898, -80.7808): {361260: 3}, (35.7235, -79.4254): {361260: 1}, (36.1658, -80.3695): {361260: 1}, (34.1827, -79.7827): {361260: 1}, (35.4118, -80.6468): {361260: 1}, (35.3144, -80.8051): {361260: 4}, (35.908, -79.9801): {361260: 1}, (35.863, -78.5356): {361260: 3}, (35.7125, -78.8235): {361260: 1}, ...}

Table 6 The account has a discernible base and only 3 devices were used for streaming. It was accessed from a wide range of places and some device usage might be worth investigating. It is a case with some uncertainty that needs further examination

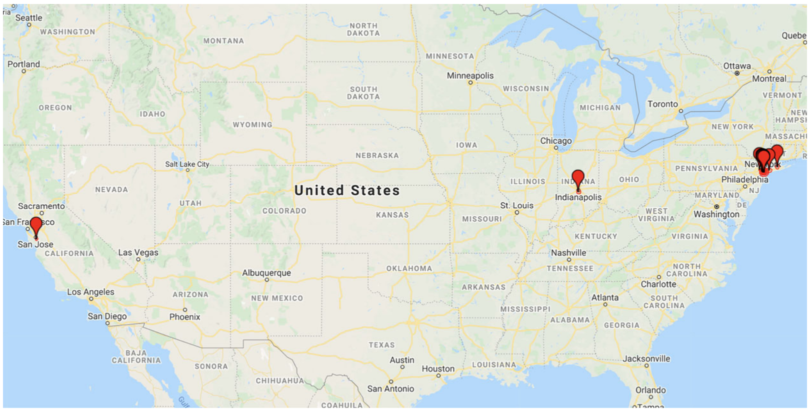
Usage visualization	
Location usage map	{ (40.6844, -73.9804): {102246: 15}, (40.703, -73.9432): {1858906: 4, 102246: 7}, (40.728, -73.9493): {1858906: 6}, (40.7155, -73.9884): {102246: 3}, (40.7324, -73.9895): {77: 1, 102246: 6}, (40.72, -74.0037): {1858906: 1}, (40.6401, -73.9044): {1858906: 3}, (40.7141, -73.9524): {102246: 7}, (40.7573, -73.5775): {1858906: 4, 102246: 2}, (40.9253, -73.0478): {1858906: 3}, **(40.6715, -73.9245): {1858906: 3, 77: 2, 102246: 25}** , (40.6693, -73.8955): {1858906: 11, 102246: 1}, (40.6797, -73.9503): {1858906: 5}, (39.6748, -86.1277): {102246: 4}, ...}

Table 7 A typical shared account: used in many locations, without a clear base location (many locations have similar numbers of usages); 39 devices have been used for streaming with this account

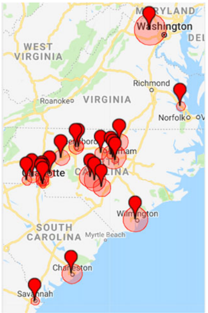
Usage visualization	Location usage map
	(35.2469, -81.3611): {1167950: 5}, (35.3279, -81.1805): {1167950: 8}, (36.2232, -78.4402): {226001: 5, 714995: 8, 8152244: 4, 1416869: 2, 54: 1, 227271: 7}, (35.8417, -78.6325): {611569: 5, 7923074: 31, 6138686: 3, 1167950: 38, 3143206: 9, 2024894: 20, 4657176: 2, 6166105: 4, 7279927: 1, 54: 6, 504542: 1}, (35.3413, -79.3625): {226001: 7, 714995: 32, 1416869: 35, 54: 3, 227271: 4, 6806357: 2, 3062359: 1, 1655: 2, 4759605: 1}, (35.2862, -80.8798): {1167950: 10}, (35.1331, -80.8597): {384645: 1, 54: 1}, (35.2285, -80.8449): {6166105: 2, 1167950: 5}, (35.2427, -79.2277): {226001: 2, 714995: 43, 2879204: 13, 1416869: 19, 54: 1, 227271: 1}, ...

Table 8 A definitely shared account: used in numerous locations with 200+ devices being used for streaming with this account in the 3-month period

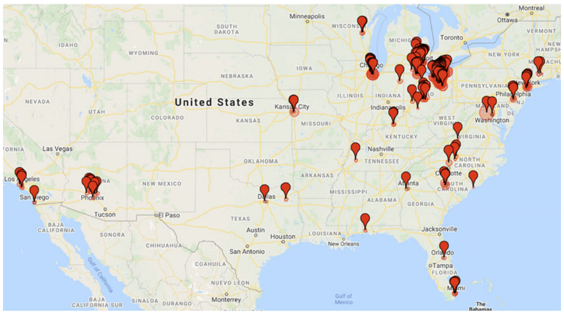
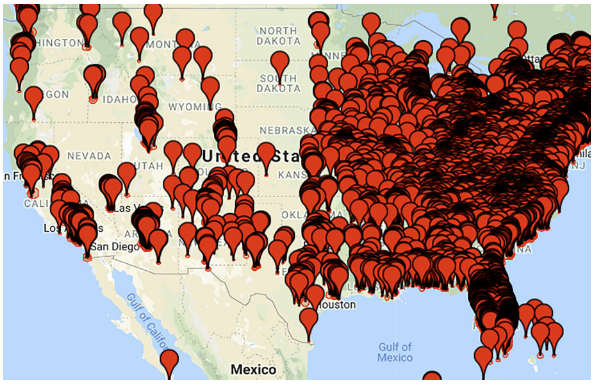
Usage visualization	Location usage map
	The list of locations and devices is too long to put here.

Table 9 A wildly shared account: used in numerous locations; 33,909 devices have been used for streaming with this account in the 3-month period. This is potentially a fraudulent sharing. It is unlikely that the owner would share with so many people

Usage visualization	Location usage map
 NORTH DAKOTA SOUTH DAKOTA WYOMING IDAHO NEBRASKA NEVADA UTAH United States San Diego Houston Gulf of California Gulf of Mexico Mexico	The list of locations and devices is too long to put here.

has been a design principle for our solution. As we have demonstrated, our identification results can be illustrated intuitively and digested easily. This surely helps our solution and results to be more trustworthy.

All algorithms in our solution are naturally parallelizable: we can easily split the computation by grouping account userIDs. In our implementation, we divide the work by the first character of userIds, i.e., [0-9, a-f], so the work was split into 16 batches. Thus we don't need to have a huge hashmap of all users, instead we work with 1/16 of them at each batch. This greatly reduced the memory requirement for our implementation. We use only one workstation for this experiment. It can finish all jobs in a week, which is enough for (the currently designed) monthly sharing score update. In the future, we can use a machine cluster to scale for more users if necessary.

As our solution is based on GPS coordinates, it is possible that in high population density areas, e.g. high rise apartments, people can share their accounts without being caught, since they are indistinguishable by position alone. The fact that we also consider the number of devices mitigates this to some extent. Nevertheless, more information such as the number of concurrent sessions and user behavior analysis will be needed to better address the issue. In any case, the fact that we can identify over 6% of accounts as reliably shared accounts can already have a significant impact, potentially saving service providers hundreds of millions of dollars.

Since we can identify millions of sharing accounts (unauthorized) with high confidence, it also opens up possibility of doing more user behavior based analysis. As mentioned Section 2, there is significant amount of existing work on identifying multiple users of one account. Nevertheless, none of them worked on identifying legitimate vs. unauthorized shared accounts, because there is no human labeled data to learn from. Based on our results we can create a pseudo-labeled data set. E.g., using the top 5% high risk accounts as one class and the accounts with the bottom 70% score as the negative class, the remaining 25% accounts as undecided. Then we can analyze different usage patterns between unauthorized and legitimately shared accounts. For example, we can learn to classify shared accounts using some supervised learning algorithm, e.g., one class SVM [6, 12] and more recently XGBoost [23]. Then we can use the learned model to help determining the labels for the 25% undecided accounts, in conjunction with the score estimated from Algorithm 3. The pseudo-labeled data will be imbalanced as only 5% data are labeled as positive. For better model quality, the data can be balanced using sampling approaches [14, 21].

In our TV-everywhere dataset which has about 30 million users, 1.98 million accounts are identified as being shared. Even if only 10% of these shared users are converted to regular accounts, that is nearly 200k new accounts. If they all choose the low tier service with monthly subscription cost of 10 dollars, that leads to $\$2M$ extra revenue each month and over $\$20M$ each year. It is possible that multiple friends put their money together to pay for one subscription. So there is a chance that an individual cannot afford the subscription without sharing. So they have to quit, which leads to revenue loss. Nevertheless, this is very unlikely as the cost is fairly small. In addition, even in the extreme case that these shared users quit due to affordability, their cancellation lead to reduced resource consumption (server and network infrastructure) to the service provider, thus lowering their infrastructure cost. Note these shared users usually take much more resources than regular users because they include more families. So their cost to service providers are higher, although they are paying the same monthly fee as regular users. Nowadays, infrastructure is a major factor in business operation. Therefore, losing a shared account, although not desirable, is not as bad as losing a regular customer. The growth from conversion of shared accounts to regular account will greatly outweigh the loss of shared accounts.

A byproduct of our solution is that we can find some interesting group of users, which could open up new opportunities for service providers. For example, we found a significant number of *traveling accounts*: legitimate users traveling frequently such as the cases in Tables 4 and 5. This offers service providers an opportunity to create value-added services specially targeting these group of users, thus bringing in more revenue.

5 Conclusion

Subscription-based business model is becoming more and more popular. Many industry giants such as Microsoft, Amazon and Apple have launched subscription based services. Video and music streaming service providers like Netflix and Disney have seen substantial growth in their subscription business, even during the COVID-19 pandemic when other businesses are shrinking. While enjoying their growth, service providers are also facing the problem of unauthorized sharing of subscriptions (to users outside of owners' household). According to multiple studies, account sharing is common among subscribers of video streaming services, which leads to huge revenue loss for service providers. In order to address this problem, we propose a novel solution for identifying shared accounts for streaming services. The solution is very efficient; a single machine can process 3 months of data with 30 million users in a week. We can reliably identify over 2 million shared accounts which means potential revenue of hundreds of millions of dollars. Our results are *explainable*. Each account gets an intuitive and interactive web-based visualization, so service providers can understand why the account is labeled as normal or unauthorized sharing. The proposed solution tolerates noise in geo-locations, so variations in GPS coordinates will not generate false alerts. Our solution also guards against geo-spoofing, making it hard for subscribers to circumvent the check. Lastly, user privacy is meticulously preserved: the users and their devices are obfuscated with integers when showing results, so service providers do not need to worry about violation of privacy. Although the proposed solution is only tested with data from the TV Everywhere ecosystem, it can be easily applied to other video streaming services such as Netflix and Hulu, as well as more general subscription-based services as long as the usage data is available. In addition, once verified with human interaction, our results can serve as ground truth labels. This will makes it possible to apply more machine learning techniques to tackle the problem. We believe the proposed solution is an important step towards solving the problem of unauthorized account sharing.

References

1. Thomson reuters poll (2014) https://fingfx.thomsonreuters.com/gfx/rngs/USA-TELEVISION-PASSWORDS-POLL/010041YS48H/index.html
2. Consumer reports poll (2015) https://www.consumerreports.org/cro/magazine/2015/01/share-logins-streaming-services/index.htm
3. Gps accuracy (2019) https://www.gps.gov/systems/gps/performance/accuracy/
4. Bajaj P, Shekhar S (2016) Experience individualization on online tv platforms through persona-based account decomposition. In: 24Th ACM international conference on multimedia, ACM, pp 252–256
5. Bloom D (2021) Why netflix's password sharing crackdown makes sense in today's streaming world. https://www.forbes.com/sites/dbloom/2021/03/16/why-netflixs-password-sharing-crackdown-makes-sense-in-todays-streaming-world
6. Dreiseitl S, Binder M (2010) Outlier detection with one-class svms: an application to melanoma prognosis. In: AMIA Annual symposium proceedings
7. Farrell M (2018) Top 25 mvpds. https://www.multichannel.com/news/top-25-mvpds-411157
8. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976
9. Godwin C (2021) Netflix is testing a crackdown on password sharing. https://www.bbc.com/news/technology-56368698
10. Hoffman C (2020) How to stop your disney+ account from getting hacked. https://www.howtogeek.com/448871/how-to-stop-your-disney-account-from-getting-hacked/
11. Jiang JY, Li CT, Chen Y, Wang W (2018) Identifying users behind shared accounts in online streaming services. In: The 41st international ACM SIGIR conference on research & development in information retrieval, ACM, pp 65–74
12. Jordaan EM (2004) Robust outlier detection using svm regression. In: International joint conference on neural networks
13. Kaufman D (2019) Synamedia offers ai solution to password sharing. https://www.etcentric.org/ces-2019-synamedia-offers-ai-solution-to-password-sharing/
14. Kubat M, Matwin S et al (1997) Addressing the curse of imbalanced training sets: one-sided selection. ICML, pp 179–186
15. Morris S (2019) Netflix password sharing crackdown after users swapping accounts loses streaming giant 135 million a month. https://www.newsweek.com/netflix-password-sharing-october-2019-1466711
16. Salinas S (2018) Millennials are going to extreme lengths to share streaming passwords. https://www.cnbc.com
17. Sherman A (2021) Netflix crackdown on password sharing. https://www.cnbc.com/2021/03/11/netflix-password-sharing-crackdown-being-tested.html
18. Verstrepen K, Goethals B (2015) Top-n recommendation for shared accounts. In: 9Th ACM conference on recommender systems, ACM, pp 59–66
19. Wang Z, Yang Y, He L, Gu J (2014) User identification within a shared account: Improving ip-tv recommender performance. In: East european conference on advances in databases and information systems, Springer, pp 219–233
20. Zhang A, Fawaz N, Ioannidis S, Montanari A (2012) Guess who rated this movie: Identifying users through subspace clustering. In: Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence

21. Zhang W, Kobeissi S, Tomko S, Challis C (2017) Adaptive sampling scheme for learning in severely imbalanced large scale data. In: Asian conference on machine learning, pp 240–247
22. Zhao Y, Cao J, Tan Y (2016) Passenger prediction in shared accounts for flight service recommendation. In: Asia-pacific services computing conference, Springer, pp 159–172
23. Zhao Y, Hryniewicki MK (2018) Xgbod: improving supervised outlier detection with unsupervised representation learning. In: IJCNN

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Wei Zhang is a Senior Staff Machine Machine Learning Scientist in Adobe. He has extensive experiences in research and development in machine learning and data analytics. He author/co-authored over 20 peer reviewed papers on top AI/ML conferences and journals with 2000 citations. He also got 12 granted patents and 8 more pending approval (most as the first inventor).