



# Automatic Identification of Account Sharing for Video Streaming Services

Wei Zhang<sup>(✉)</sup> and Chris Challis

Adobe Inc., McLean, USA  
wzhang@adobe.com

**Abstract.** According to multiple studies, account sharing is common among subscribers of video streaming services. This leads to huge revenue loss for service providers. Although they have strong financial interests to address the problem, service providers face multiple challenges when trying to identify shared accounts. On one hand, the huge volume of unstructured and noisy data makes it hard to manually process data. On the other hand, it is legitimate for family members to share an account, from anywhere and use many devices as they want. Only these accounts which are shared outside of the household are against policies. In this paper, we propose an efficient solution to address the account sharing problem. Based on a massive volume of session data, our solution builds user profile through accumulating and representing the geolocation and device usage information. Then we estimate the account sharing risk by analyzing the usage pattern of each account. The proposed solution can identify a significant number of shared accounts and help service providers to recoup a huge amount of lost revenue.

## 1 Introduction

Video streaming is a massive industry and keeps growing. Account sharing is a major problem faced by streaming service providers. According to a poll by Consumer Reports in 2015 [2], 46% of respondents who use a streaming service share their account with someone outside their households. An earlier poll by Thomson Reuters in 2014 found that 15% to 20% millennials shared their accounts [1]. More recent, a study by CNBC in 2018 found that an estimated 35% of millennials share passwords for streaming services [7]. Consequently, the streaming industry loses huge potential revenue due to account sharing. The loss could be hundreds of millions of dollars annually for Netflix alone [7].

Although streaming service providers have strong financial interests in addressing the problem, they face multiple challenges when trying to identify shared accounts. 1. Huge volume of data: leading providers have millions of users and billions of sessions each month. 2. Unstructured and noisy data: session logs are plain text with lots of noise: missing information, numerical error, etc. 3. Perhaps the trickiest problem is that it is perfectly legitimate to share within a household. Family members can share one account from anywhere (e.g. home, office, school or on travel) and use any device. Only accounts which are shared across household are against policies and should be pursued.

Currently, service providers choose not to manually identify/label account sharing because it is too costly and prone to error. Note *account sharing* in this paper refers to agnost-policy sharing unless we explicitly note otherwise.

## 1.1 Problem Definition

In this paper, we focus on the video streaming service in the TV Everywhere ecosystem, although our solution can be easily adapted to other services. TV Everywhere is also known as authenticated streaming service, 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 [5]. 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.

```
2018-04-02 01:00:55.911 [cex-38] INFO com.adobe.tv.metrics.MetricsLogger.onlyMetrics - [METRICS]
event=authzgmvpd=Charter_Direct&prog=ESPN&uid=130d92fdc8e2eb8e906dfafe93b0785f&plainId=AQICi2beoJu07syR9xJRvCcxTBq
PPZMKuQymHesNIM0bE17BoGIuTly5KWPE1Vr8Xg3qEWI Ewdj XM%3D&ip=174.110.163.2&deviceId=%3CsimpleTokenFingerprint+xmlns%3D
%22http%3A%2F%2Ftve.adobe.com%2Fsd%2Ftokens%2Fsimple%22%3Eba0458ac546100d7259410da848aa4321ac62875%3C%2FsimpleToken
Fingerprint%3E&arch=3.0&latency=325&res=%3Crss+version%3D%22.0%22+xmlns%3Amedia%3D%22http%3A%2F%2Fsearch.yahoo.com%
2Fmss%2F%22%3E%3Cchannel%3E%3Ctitle%3E%3C%21%5BCDATA%5Bspn1%5D%3E%3C%2Ftitle%3E%3Citem%3E%3Ctitle%3E%3C%21%5BCD
ATA%5BSt.+Louis+Cardinals+vs.+NewYork+Mets+%28re-
air%29%5D%3E%3C%2Ftitle%3E%3Cguid%3E%3C%21%5BCDATA%5Bspn1%2Fst.+Louis+Cardinals+vs.+NewYork+Mets+%28re-
air%29%2F3292023%5D%3E%3C%2Fguid%3E%3Cmedia%3Arating+scheme%3D%22urn%3Av-
chip%22%3E%3C%21%5BCDATA%5BG%5D%3E%3C%2Fmedia%3Arating%3E%3C%2Fitem%3E%3C%2Fchannel%3E%3C%2Frss%3E&tll=86400&devi
ce=dnr&clientType=ClientLess&clientVersion=v1&os=RokuOS&pht=SetTopBox&dmd=Digital+Video+player&dhwnd=Digital+Video+p
layer&dhvvn=Roku&dhwmf=Roku&dosnm=Roku+OS&dosfm=Roku+OS&dosvn=Roku&ddsw=0&ddsh=0&ddsp=0&userAgent=Roku%2FDVP-
8.0+%28288.00E94128A%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 information such as the subscriber’s ID (obfuscated), the device used for connection, the GPS location, etc.

We propose to utilize the session information for creating a service which automatically identifies shared accounts. MVPDs can benefit from the service in two ways: (1) the opportunity of growing revenue remarkably: conversion of shared accounts to regular paid accounts, even just a small fraction of them, will bring in millions of dollars because of their large customer bases; (2) limiting the number of shared accounts also translates to server/network load reduction, which leads to significant cost cutting.

One big challenge for this service is that there is no ground truth label, since the labeling cost is prohibitive as we discussed in Sect. 1. This certainly constrains our choice of algorithms. More importantly, a big question arises: without any ground truth to compare against, how can we justify the results of the service? This question is *crucial* for demonstrating the value of the service due to the consequences of regulating account sharing: treating normal accounts as shared accounts (false alarm) will annoy their subscribers and lead to potential loss of business. Classifying shared accounts as regular accounts, on the other hand, means the solution brings no value to them. 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 MVPDs can easily understand the reason so to trust the results.

## 2 Existing Work

The streaming industry has tried to restrain account sharing by adding constraints to user accounts: (1) limiting the number of concurrent streaming; (2) asking users to register a limited number of devices to their accounts. However, the first approach can adversely affect concurrent streaming by family members, who are entitled to do so. In addition, limiting concurrent streaming can be circumvented by sharing accounts at different time periods. Having to register a limited number of devices will hurt customer experience and is not desirable either: first, it is a hassle to do the registration; second, customers are having more and more devices and they can easily hit the limit. In addition, an account owner can sell/give the “registered” device to other people, so they can use the device for streaming and easily defeat the policy.

In the academic community, there has been considerable research [4,8–11] on modeling user behavior from session logs, mainly for improving recommendations. They mostly focus on identifying multiple users by the content that they watched. The techniques that were used are: collaborative filtering [8], subspace clustering [10], graph partitions [9] and topic modeling [11]. [6,10] are the very few paper which attempt to determine whether an account is shared by multiple users. However, multiple users sharing one account are not necessarily against-policy. In fact, more often than not, they are shared within a household. Therefore, they cannot solve the account sharing problem.

## 3 Our Solution

The account sharing detection service 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 MVPDs to maintaining and growing 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 stream video anywhere they want.

Although the problem is complicated, the following assumptions usually hold. (1). Even though they share the account, users outside of the household (against-policy sharing) are unlikely to share devices with account holders, since they live in different places. They might use devices which used to be owned by the account holder, through sale/gift, but the devices are transferred and not shared, i.e., the account owner is unlikely to use it again. (2). Non-family users are likely to stream videos from separate locations, not the home of account holders.

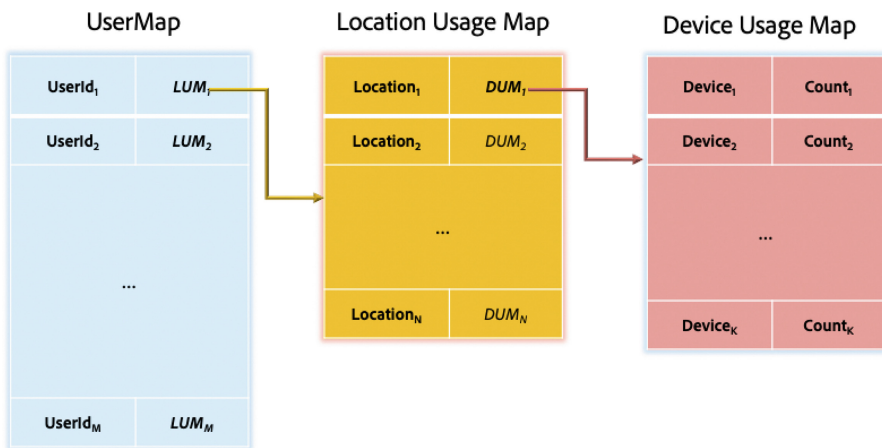


Fig. 2. The data structures used in the algorithms.

Otherwise, they are more like a part of the household and virtually impossible to be identified. Based on these analysis, we propose an novel approach to estimate a sharing score of each account. It utilizes both geolocation and device information to address the problem. Algorithm 2 describes how the sharing score is estimated. It depends on Algorithm 1 to process data and construct efficient retrievable user profiles. Note the GPS coordinates in the session logs are usually noisy; we need to mitigate the problem when processing raw log files as described in Algorithm 1. Without this process, multiple geolocations might be associated with an account even if the owner only streams in her/his home, because the GPS coordinates of difference sessions can be different due to noise.

We first introduce the notation and data structures shown in Fig. 2.

1. *Device Usage Map (DUM)* represents a distribution of device usages. It is a hashmap  $\mathcal{M} = \{(\mathcal{D}^0 : \mathcal{C}^0), (\mathcal{D}^1 : \mathcal{C}^1), \dots, (\mathcal{D}^K : \mathcal{C}^K)\}$ ,  $K$  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 usages for the  $k^{th}$  device.
2. *Location Usage Map (LUM)* is a hashmap in the form  $\{(\mathcal{L}_0 : \mathcal{M}_0), (\mathcal{L}_1 : \mathcal{M}_1), \dots, (\mathcal{L}_N : \mathcal{M}_N)\}$ , 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:  $\{(40.2814, -111.698) : \{(4277780 : 16), (11090085 : 2)\}\}$ . It means that 2 devices have been used at location (40.2814, -111.698): device #4277780 was used 16 times (appears in 16 sessions), device #11090085 was used twice.
3. *userMap* stores the profile of all users; each entry contains an userID and a list of *Location Usage Maps* 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 *LUMs* 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 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:  $\{ (\text{original deviceID} : \text{device index}), \dots \}$ . It represents a mapping from the original deviceID (a string) to a integer value, e.g., 4277780 in the above example. Checking whether a device exist in the system is super efficient using hashmap. In addition, a device can appear in many locations, even across users. Using integer instead of string can reduce the space requirement significantly.

---

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

---

1. Initialization:  $\text{deviceMap} \leftarrow \{\}, \text{userMap} \leftarrow \{\}, n\text{Devices} \leftarrow 0$
  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  $\in$  deviceMap **then**  
     Get its device index  $k$ .  
   **else**  
     Add  $\{ \text{deviceID} : n\text{Devices} \}$  to deviceMap  
      $k \leftarrow n\text{Devices}, n\text{Devices} \leftarrow n\text{Devices} + 1$
    - (c) **if** userID  $\in$  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  $\mathcal{M}_j$   
     **else**  
       Add  $\{ \mathcal{X} : \{k : 1\} \}$  to the *LUM*  
   **else**  
     Create an entry in userMap for this new account
  3. Go through each user in the userMap, combine neighboring *Location Usage Maps*:
    - (a) Sort *LUMs* by their number of device usages, in descending order.
    - (b)  $i \leftarrow 0, P \leftarrow$  number of *LUMs*  
   **while**  $i \neq P$  **do**  
     **for all**  $(\mathcal{L}_j : \mathcal{M}_j), j > i$  **do**  
       **if**  $\text{dist}(\mathcal{L}_i : \mathcal{L}_j) < \sigma$  **then**  
         Merge  $\mathcal{M}_j$  into  $\mathcal{M}_i$  (combine their device usage)  
         Delete  $(\mathcal{L}_j : \mathcal{M}_j)$   
          $P \leftarrow P - 1$   
      $i \leftarrow i + 1$
-

Assuming the GPS noise follows a zero mean distribution, the observed coordinates will be centered around the true coordinates. The step 3 in Algorithm 1 is essentially doing non-maximum suppression, it combines neighboring Location Usage Maps into one single usage map in the center location<sup>1</sup>.  $\sigma$  is set to be 50 m in our experiment. The average GPS accuracy is about 7.8 m [3], well within the  $\sigma$  range. So we can almost guarantee to handle GPS noise. This  $\sigma$  setting is also fine enough to identify account sharing across street. If the GPS noise do not follow a zero-mean distribution, the coordinates will be shifted by the non-zero mean. However, it will not affect the scoring algorithm since Algorithm 2 is only based on the usage pattern, not the exact location.

---

**Algorithm 2.** Sharing score estimation for each user in userMap.

---

1. Given  $[(\mathcal{L}_0 : \mathcal{M}_0), (\mathcal{L}_1 : \mathcal{M}_1), \dots, (\mathcal{L}_N : \mathcal{M}_N)]$ , 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  $i = 1, 2, \dots, N$ :
    - (a) Calculate distance weight:  $W_d \leftarrow \log_\alpha(\max(\text{dist}, \alpha))$ , where  $\text{dist} \leftarrow |\mathcal{L}_i - \mathcal{L}|$ .
    - (b) Let  $\mathcal{M}_i$  be  $\{(\mathcal{D}_i^0 : \mathcal{C}_i^0), (\mathcal{D}_i^1 : \mathcal{C}_i^1), \dots, (\mathcal{D}_i^J : \mathcal{C}_i^J)\}$ , where  $J$  is the number of devices associated with the  $i^{\text{th}}$  location.  $\mathcal{D}_i^j$  is the  $j^{\text{th}}$  device ID,  $\mathcal{C}_i^j$  is the count of usages for the  $j^{\text{th}}$  device.
 

**for**  $j = 0, 1, \dots, J$  **do**

$R_j \leftarrow 0$

**if**  $\mathcal{D}_i^j \in \mathcal{M}$ , say  $\mathcal{D}_i^j = \mathcal{D}^k$  **then**

$r \leftarrow \frac{\mathcal{C}_i^j}{\mathcal{C}^k}$

$R_j \leftarrow R_j + \frac{1}{e^{-(r-\beta)/3} + 1}$

$\mathcal{C}^k \leftarrow \mathcal{C}^k + \mathcal{C}_i^j$

**else**

$R_j \leftarrow R_j + 1$

Add  $(\mathcal{D}_i^j, \mathcal{C}_i^j)$  to  $\mathcal{M}$
    - (c)  $R \leftarrow R + R_j * W_d$
  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 list of Location Usage Maps, in descending order of their device usages. We use Google Map API for visualizing the usage pattern, so people can easily see why an account is labeled as normal or abusive sharing. Some screen copies of the interactive map are shown in the result section, such as Table 1, Table 4 and Table 5. 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

<sup>1</sup> In our implementation, we use kernel density estimation to in stead of the simple histogram count. So it will be more robust to GPS drift.

the circle, the larger the number of usages (sum of all device usages in that locations). Note the circle size is not linearly proportional to the number of usages. Because the range of usage numbers is very wide, from 1 to multiple thousands. Consequently, a large circle will make all other circles too small to see. 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 usages is called the *base location* of the account, presumably it is the owners home place. The base location is important for the scoring Algorithm 2. A regular account is more likely to have a dominant base location, because that is the place where most household members enjoy the streaming service. For a heavily shared account, the usage pattern is more distributed.

Algorithm 2 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 (5) in Algorithm 2. We believe that the risk of sharing is fairly low when the non-base usage is not significantly higher than the usages at the base location.  $\beta$  is set to be 20 in our experiments. That is, when the usage in other locations is 20 times as high, we increase  $R_j$  by 0.5. Higher  $R_j$  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 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 office 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  $\alpha$  is set to be 50 so the distance weight will have no effect in Algorithm 2(3c) for usages within 50 miles, while usage in locations which are hundreds of miles away will be penalized and lead to higher scores.

Even if all streaming 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 2 is designed for this purpose. The higher the number of devices, the higher the weight. So even if all steaming 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 tackle 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 Maps* 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  $\gamma$ , the weight

is always 1. For accounts with just one location and less than 20 devices, their  $R$  value in Algorithm 2(5) are always 1. Consequently, they get score 0, which signifies 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 an shared account using geo-spoofing. It gets a score of 0.75, not extreme but high enough to be identified.

## 4 Experiments Result

We use a three-month session log of TV everywhere system for testing 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 terabyte. Using 0.5 as the threshold for sharing score, we identified about 6.45% accounts as shared, with very high confidence. Those accounts are usually quite obvious to be shared as what we show in Table 4 and Table 5. Using a threshold of 0.05, we identified about 15.66% accounts as shared. Some manual verification might be need for the these results. Nevertheless, based on random check, we can see the identified accounts are indeed likely to have been shared. 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.

### 4.1 Multiple Devices in One Location


Two cases are shown in Table 1 and 2. Although both have only one location for all sessions, the case in Table 2 is more likely to be a shared account due to an extraordinary number of devices being used. The accounts which have zero sharing scores all have the same visualization (one location with few devices).

### 4.2 Multiple Devices Multiple Locations

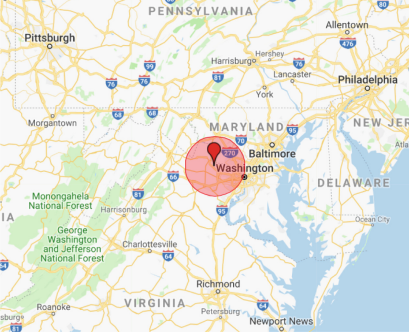
The case in Table 3 represents an interesting pattern: many locations and few devices. We call it a traveling account. The account has two devices (#747409 and #868586) associated with it. The base location is (40.7046, -73.9216) where device #747409 was used 15 times and device #868586 was used 6 times. Both devices were used in other locations, suggesting that they were taken to travel around. The account sharing score is relatively 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



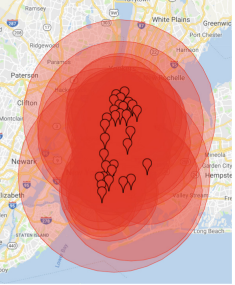
**Table 1.** This account has been used by 27 devices in total. However, they are all used in the base location and the number of devices is not super high, so it is still likely a legitimated account with score 0.05.

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

**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 might be used here. The score is 0.21.

Usage visualization	Location Usage Map
	$\{(39.0329, -77.4866): \{6435225: 4, 7796609: 2, 13210498: 2, 5182979: 4, 8522116: 2, 8911041: 3, 11916204: 2, 12181898: 1, 13113230: 2, 12885648: 2, 6745537: 2, 9590035: 1, 9984404: 2, 6644503: 1, 7834904: 2, 9573145: 2, 8482497: 2, 2710447: 2, 9993242: 2, 7778113: 2, 7055009: 3, 13225123: 2, 8222628: 1, 6520613: 2, 7564072: 3, 3754666: 2, 7653420: 2, 11134208: 2, 8911023: 2, 2710448: 2, 12746418: 2, 12634526: 2, 7379895: 4, \dots \}\}$

**Table 3.** This account has been used in many locations. The location with most usage is labeled bold. However, the account is identified as a regular account because few devices are used. It is most likely family members traveling around.

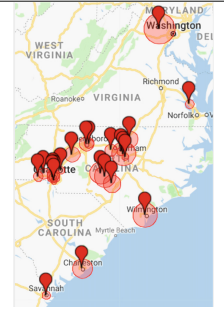
Usage visualization	Location Usage Map
	$\{(40.6797, -73.9503): \{747409: 2\}, (40.859, -73.8908): \{747409: 1\}, (40.8371, -73.8807): \{868586: 1\}, (40.679, -73.9618): \{747409: 1\}, (40.809, -73.9168): \{868586: 1\}, (40.8276, -73.896): \{747409: 1\}, (40.7046, -73.9216): \{747409: 15, 868586: 6\},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

higher according to Algorithm 2, 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 account holders' base location, in order to "register" their devices to the account. Otherwise, they will be identified.


### 4.3 Identified Shared Accounts

See Table 4 and Table 5 for some typical cases.

**Table 4.** 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
	<p>(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}, ...</p>

**Table 5.** A wildly shared account: used in numerous locations; 33,909 devices have been used for streaming with this account in the 3-month period.

Usage visualization	Location Usage Map
	<p>The list of locations and devices is too long to put here.</p>

#### 4.4 Discussion

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

Both Algorithm 1 and Algorithm 2 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, e.g., 16 machines to process the 16 batches.

Since our solution is based on GPS coordinates, it is possible that in high population density areas, e.g. high rise apartment, 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% accounts as reliably shared accounts can already have a significant impact, potentially saving streaming service providers hundreds of millions of dollars.

## 5 Conclusion

In this paper, we have proposed a novel solution for identifying shared accounts for video streaming services. It has several major advantages. First, it is efficient; we can process 3 months of data with 30 million users in a week using one single machine, with over 2 million shared accounts detected. Second, the results are *explainable*. Each processed subscriber, whether labeled as shared or regular, is giving an intuitive and interactive web-based illustration, so that service providers can understand and trust the results. Note that our solution preserves privacy: we obfuscate deviceID using an integer when showing the result, so service providers do not need to worry about violation of privacy when using our solution. In addition, the proposed solution handles noise in geolocation information. Last but not the least, it guards against geo-spoofing, making it hard for subscribers to game the system.

Although our solution is designed in the context of TV Everywhere ecosystem, it can be directly applied to other video streaming services such as Netflix and Hulu. In addition, it can be generalized to other applications, e.g., music streaming and more broadly, subscription-based software, e.g., Photoshop.

## 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.: Experience individualization on online TV platforms through persona-based account decomposition. In: 24th ACM International Conference on Multimedia, pp. 252–256. ACM (2016)
5. Farrell, M.: Top 25 MVPDs (2018). <https://www.multichannel.com/news/top-25-mvpds-411157>
6. Jiang, J.Y., Li, C.T., Chen, Y., Wang, W.: Identifying users behind shared accounts in online streaming services. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 65–74. ACM (2018)
7. Salinas, S.: Millennials are going to extreme lengths to share streaming passwords (2018). <https://www.cnbc.com>
8. Verstrepen, K., Goethals, B.: Top-n recommendation for shared accounts. In: 9th ACM Conference on Recommender Systems, pp. 59–66. ACM (2015)
9. Wang, Z., Yang, Y., He, L., Gu, J.: User identification within a shared account: improving IP-TV recommender performance. In: Manolopoulos, Y., Trajcevski, G., Kon-Popovska, M. (eds.) ADBIS 2014. LNCS, vol. 8716, pp. 219–233. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10933-6\\_17](https://doi.org/10.1007/978-3-319-10933-6_17)
10. Zhang, A., Fawaz, N., Ioannidis, S., Montanari, A.: Guess who rated this movie: identifying users through subspace clustering. In: Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (2012)
11. Zhao, Y., Cao, J., Tan, Y.: Passenger prediction in shared accounts for flight service recommendation. In: Wang, G., Han, Y., Martínez Pérez, G. (eds.) APSCC 2016. LNCS, vol. 10065, pp. 159–172. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-49178-3\\_12](https://doi.org/10.1007/978-3-319-49178-3_12)