Behavior Analysis in the Medical Sector: Theory and Practice

Mahdi Alizadeh
Eindhoven University of Technology
m.alizadeh@tue.nl

Sander Peters
Eindhoven University of Technology
s.p.c.peters@student.tue.nl

Sandro Etalle
Eindhoven University of Technology
s.etalle@tue.nl

Nicola Zannone
Eindhoven University of Technology
n.zannone@tue.nl

ABSTRACT

Behavior analysis has received considerable attention over recent years. In this paper, we apply behavior analysis to study the use of the Break-The-Glass (BTG) procedure at the Academic Medical Center (AMC), a large Dutch hospital. Similar to most hospitals, AMC employs the BTG procedure to deal with emergencies, which allows users to access patient data that they would not be normally allowed to access. This flexibility can be misused by users, leading to legal and financial consequences for the hospital. To assist AMC in the detection of possible misuses of the BTG procedure, in this work, we present an approach to analyze user behavior and apply it to a log collected from AMC. We partition users into different subgroups and build self-explanatory histogram-based profiles for users and subgroups. By comparing profiles, we measure to what extent users behave differently from their peers. The discussion of our findings with experts at AMC has shown that our approach can provide meaningful insights on user behavior and histograms are easy to understand and facilitate the investigation of suspicious behaviors.

CCS CONCEPTS

• **Security and privacy** → **Database activity monitoring**; • **Applied computing** → *Investigation techniques*;

KEYWORDS

Behavior analysis, break-the-glass, healthcare

1 INTRODUCTION

Nowadays, many organizations log users' interaction with their IT systems. These logs can be analyzed for understanding user behavior and the obtained insights can serve several purposes. For example, they can be used to identify users misusing the system or to enhance the employed policies. In this paper, we apply behavior analysis to study the use of the Break-The-Glass (BTG) procedure at the Academic Medical Center (AMC). AMC is one of the largest hospitals in the Netherlands and, similar to other healthcare organizations, collects large amounts of patient information, including demographics, medical history, laboratory test results and billing information. Given the high sensitivity of medical information, its protection against data breaches and other threats is a main concern for AMC.

According to a study conducted by the Ponemon Institute in 91 healthcare organizations and their 84 business associates in 2016 [19], data breaches have increased in frequency and cost. Among healthcare providers represented in this study, nearly 90% of them had at least one data breach in the past two years and 45% had more than five data breaches. The average cost of data breaches is estimated to be more than $2.2 million per incident. In response to these risks, a number of regulations and guidelines such as the General Data Protection Regulation (GDPR) [10] have been enacted. These regulations mandate organizations to have frameworks in place for protecting patient privacy.

To comply with these regulations and protect patient privacy, AMC employs access control mechanisms to determine which data a user can access. Traditional access control mechanisms, when correctly deployed, can provide theoretical guarantees that unauthorized accesses are prevented. However, they are too inflexible to be used in dynamic environments like AMC, where exceptions and unpredictable circumstances often arise. For example, to deal with emergencies when the treating doctors are not available, other doctors might need to access patient data. Preventing such accesses can disrupt patient care and have fatal consequences for patients.

Thus, alongside access control mechanisms, AMC employs mechanisms like the BTG procedure that allows users to bypass preventive enforcement mechanisms. This flexibility, however, introduces a weak point in the system that can be misused by users. For example, a user may use the BTG procedure to collect patient information and disclose it to outsiders for profit or revenge [4]. To this end, user actions are recorded in logs and later analyzed by the privacy and security officers at the hospital to detect possible data misuses.

In AMC's current practices, logs are analyzed manually based on textual reports. This analysis, however, is time consuming, inefficient and costly, making it impossible to react in a timely manner to reduce the risk of user actions. In fact, thousands of data accesses and invocations of the BTG procedure are recorded every day, resulting in a large amount of logs to be analyzed. Thus, a (semi)automatic approach is needed to support analysts in the investigation of logs.

In this paper, we present an analysis of the use of the BTG procedure at AMC and identify possible misuses of the procedure. Our contributions can be summarized as follows:

- We propose an approach to assist analysts in the analysis of user behavior and in the detection of attacks spanned over multiple actions. We partition users into groups based on their role and

SAC 2018, April 9–13, 2018, Pau, France

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167307

build histogram-based profiles representing user and group behavior. By comparing profiles, we measure to what extent a user behaves differently from users having the same job functions.
- Our approach uses off-the-shelf clustering algorithms to refine group profiles. Specifically, these algorithms make it possible to group together users that exhibit similar behavior. This leads to the construction of more accurate profiles, thus enhancing detection capabilities.
- Our approach makes use of histograms to represent profiles. A discussion of our findings with experts at AMC has shown that this representation is easy to understand and facilitates the analysis of suspicious behavior.
- We demonstrate how data analysis techniques can be employed to better understand logs and derive the actual policies implemented within the hospital. This analysis can also help selecting the features to be used for clustering and behavior analysis.
- We implemented our approach as a plug-in for RapidMiner, a software platform for data science, and evaluated its applicability to real-life scenarios using a log collected from AMC.

The remainder of the paper is organized as follows. The next section introduces our case study. Section 3 presents our approach for analyzing user behavior. Section 4 presents an analysis of the dataset. Experimental results are presented in Section 5. Finally, Section 6 discusses related work, and Section 7 concludes the paper and provides directions for future work.

2 CASE STUDY: AMC

The Academic Medical Center (AMC) is the hospital affiliated with the University of Amsterdam. It provides healthcare services, it hosts research activities, and it educates and assists medical students. AMC is organized in divisions, consisting of various departments offering specialized healthcare services. For example, the division responsible for child healthcare comprises the pediatrics and surgery departments. Hospital personnel are assigned a role based on their job function (e.g., doctor, nurse, receptionist). Personnel can be affiliated with multiple divisions/departments.

When admitted at the hospital, patients are registered at one of the departments and their information (e.g., medical history, prescriptions) is stored in medical records. These records contain sensitive information about patients and, thus, need to be protected. Like other hospitals, AMC has in place policies and practices to protect patient privacy. In our study, we interviewed various stakeholders at AMC to elicit the policies and practices being implemented. Next, we report the policies that were initially elicited based on these interviews. Then, in Section 4 we discuss how an analysis of the recorded data accesses allowed us to refine these policies.

To protect the privacy of VIP patients such as celebrities and politicians, AMC registers those patients under a fake name ($Policy_1$). Access decisions are made based on the department at which a user works and the department at which a patient is registered. Users can access information of patients registered at their department ($Policy_2$). To access information of patients registered at other departments, users have to use the BTG procedure, where they have to choose one of the predefined reasons (trial & research, new patient, emergency admission, or inter-colleague consultation) or fill in a reason before accessing patient information ($Policy_3$).

The adoption of the BTG procedure provides hospital staff with the flexibility necessary to access information as needed. However, this flexibility can be misused. To mitigate this risk, AMC records users' access to patient data in logs. Fig. 1 shows an excerpt of these logs. The log consists of the following attributes: *timestamp* of the event, *userID* of the user who accessed data, *patientID* of the patient whose data was accessed, *access type* indicating whether the access was denied or granted and if the BTG procedure was used, *explanation* stating the reason for access as filled by users, *IP* address of the computer from which the access was performed, *inventoryID* of the computer from which the access was performed, *computer Type* that can be desktop or laptop, *last active* indicating the last time logged in the network, *wall outlet* indicating network connection used by the computer (for instance, G6-267-A2 means G (building), 6 (floor number), 267 (room number), A2 connection ID), *division* from which the access request was performed, and the *role* of the user.

The analysis of such logs, however, is not trivial. As shown in the log provided by AMC (see Section 4 for detail), there can be thousands of invocations of the BTG procedure every day. Therefore, AMC is seeking a means to support security analysts in the investigation of user behavior. Consider, for instance, role *Administratie (AZP)*. This role is assigned to users that are responsible for the admission of patients to the hospital. Based on a log provided by AMC, there are 14 users with such a role. Despite having the same role, these users exhibit different behaviors. For example, users *50009433* and *30009868* executed many more actions than other users. In total, these two users together executed 3,210 actions (76.9% of all actions); while other users on average executed 80 actions. These two users often selected *trial & research* as the reason to access patient information (2,273 times); while other users never selected this reason when invoking the BTG procedure.

Behavior analysis has the potential to identify users acting differently from their peers. However, comparing users with heterogeneous behavior, as in the case of role *Administratie (AZP)*, can provide misleading diagnostics. Moreover, the presence of users performing a significantly larger number of accesses can bias the overall group behavior, making it difficult to identify common behaviors within the group. To enhance detection capabilities, similar users within a group should be identified and used as the baseline to analyze a user's behavior. Based on this analysis, security officers should be able to investigate *how* and *why* users behave differently from their peers. To facilitate this investigation, they should be supported in the understanding of the differences between user behaviors and in the analysis of root causes. In the next section, we present an approach for behavior analysis that addressees these issues.

3 APPROACH

The idea underlying our approach is to investigate users' behavior by constructing their behavioral profile and comparing these profiles with their expected behavior represented by the profile of the (sub)group to which they belong. Based on this analysis, a security analyst can take proper actions to mitigate the impact of possible misuses. Fig. 2 presents an overview of our approach. The first phase of the approach encompasses the preprocessing of log data (❶). In this phase, the relevant features for the analysis are extracted and, based on them, the log is preprocessed. The second phase aims to

ID	Timestamp	UserID	PatientID	Access Type	Explanation	IP	InventoryID	Computer Type	Last Active	Wall outlet	Division	Role
1	2015-05-01 08:17:11	50009433	3933061	Trial & Research	NULL	145.117. 68.134	09-020-1073	desktop	Jul 8 2015 12:50AM	PA0-188-D2	divDF	Administratie (AZP)
2	2015-05-01 08:41:29	10001042	4881659	Access without using BTG procedure	NULL	145.117. 230.77	09-020-4130	desktop	Jul 8 2015 12:50AM	H8 -198-A4	divCE	Lab+Brieven
3	2015-05-01 08:41:43	40005997	3895137	Other reasons	Behandelrelatie volgens X/Care	145.117. 138.182	09-020-3136	desktop	Jul 8 2015 12:49AM	G4 -176-A6	divG	BA-behandelend arts
4	2015-05-01 08:41:43	40005997	3895137	Access without using BTG procedure	NULL	145.117. 138.182	09-020-3136	desktop	Jul 8 2015 12:49AM	G4 -176-A6	divG	BA-behandelend arts
5	2015-05-01 08:41:45	10003356	1859858	Access without using BTG procedure	NULL	145.117. 127.254	11-036-0287	desktop	Jul 8 2015 12:49AM	F8 -129-A6	divCE	Verpleging (VPK)

Figure 1: An excerpt of the log

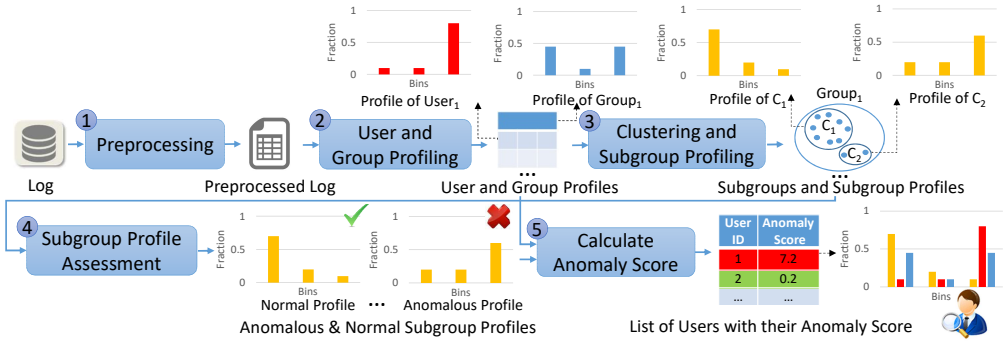

Figure 2: Overview of the approach

group users supposed to behave similarly (❷). After dividing users in groups, user and group profiles are built. In the third phase, each group is analyzed to ensure that the users in the group actually behave similarly (❸). If users have remarkably different behaviors, the group is further refined into subgroups, and subgroup profiles are constructed. In the fourth phase, subgroup profiles are analyzed and marked as *normal* or *anomalous* (❹). Finally, we compute an *anomaly score* indicating to what extent users behave differently from their peers (❺). Next, we describe each phase of the approach.

3.1 Preprocessing

User actions are typically recorded by IT systems as *events* in logs. Let $\mathcal{E}$ denotes the universe of all identifiable events. An event can be endowed with attributes that provide information about the event. Examples of these attributes can be the user that performed the action, the time when the action was performed, accessed data, etc. (see also Fig. 1). A *log* is a set of events.

The first step of our approach requires selecting features relevant to the analysis. A *feature* can be an attribute recorded in the log or can be extracted from the log to characterize relevant aspects of user behavior. In a general form, a feature can be defined as follows:

DEFINITION 1 (FEATURE). *A feature $f \in \mathcal{E} \rightarrow V_f$ is a function associating a value v from the codomain V_f to each event $e \in \mathcal{E}$. A feature value v_f is the result of applying function f to an event $e \in \mathcal{E}$, i.e. $v_f = f(e)$. A feature space $F = \langle f_1, \ldots, f_n \rangle$ is an ordered sequence of features.*

After selecting the features, the events in the log should be analyzed to extract the corresponding feature values. Based on these values, we construct an *event vector*.

DEFINITION 2 (EVENT VECTOR). *Given a feature space $F = \langle f_1, \ldots, f_n \rangle$ and an event $e \in \mathcal{E}$, an event vector $\gamma = \langle v_{f_1}, \ldots, v_{f_n} \rangle$ is a sequence of feature values v_{f_i}, where $v_{f_i} = f_i(e)$. We use Υ to denote the universe of all possible event vectors.*

Let us consider feature space $F_1 = \langle access\ type, time, division, date \rangle$. The event vector corresponding to the first event in the log of Fig. 1 is $\langle trial\ \&\ research, 08{:}17{:}11, divdf, 2015\text{-}05\text{-}01 \rangle$.

3.2 Building User and Group Profiles

In this phase, we partition users into groups and build user and group profiles. Before building these profiles, we need to partition the codomain of each feature in the feature space into a sequence of *bins*. We also define the concept of *bin frequency* to represent the frequency of the feature values that fall in a specific bin for the given event vectors. We formalize these concepts as follows:

DEFINITION 3 (BIN). *Given a feature f with the corresponding codomain V_f, a bin $b \subseteq V_f$ is a subset of V_f. A bin sequence $\beta_f = \langle b_1, \ldots, b_n \rangle$ partitions V_f into disjoint bins s.t. $\bigcup_{i=1}^{|\beta_f|} b_i = V_f$ and $\forall i \neq j, b_i \cap b_j = \emptyset$.*

DEFINITION 4 (BIN FREQUENCY, HISTOGRAM). *Given a set of event vectors $\Gamma \subseteq \Upsilon$, a feature f and the corresponding bin sequence $\beta_f = \langle b_1, \ldots, b_n \rangle$, the bin frequency for a bin $b_i \in \beta_f$ is the number of event vectors in Γ s.t. feature value $v_f \in V_f$ falls in b_i, i.e. $freq_{f,b_i}(\Gamma) = |\{\gamma | \gamma \in \Gamma \wedge v_f \in b_i\}|$. A histogram $\mu_f(\Gamma, \beta_f)$ for a feature f w.r.t. Γ and β_f is the sequence of bin frequencies, i.e. $\mu_f(\Gamma, \beta_f) = \langle freq_{f,b_1}(\Gamma), \ldots, freq_{f,b_n}(\Gamma) \rangle$.*

Note that features can have different data types. Here, we consider three data types, i.e. *nominal*, *numeric* and *time*. The partitioning of a feature value V_f may depend on its data type. For example, the feature *access type* in the log in Fig. 1 can be partitioned into its possible values, i.e. $\{new\ patient, trial\ \&\ research, \ldots\}$. However, the same approach cannot be used to partition features with

numeric and time data types. To partition these features, bins should be defined as ranges of values. For example, the *time* feature in the log in Fig. 1 can be partitioned into 12 bins with equal width (i.e. 2 hours), working ([8:00-18:00]) vs. non-working ([18:00-8:00]) hours, or working vs. non-working days. Note that bin width should be defined carefully as it can affect the accuracy of the analysis. In fact, defining many narrow bins with low bin frequency may lead to missing predominant behaviors whereas defining few wide bins with high bin frequency may lead to the inability to discriminate predominant behaviors. We discuss the features used in our analysis and binning in Section 4.

Our approach analyzes the behavior of users with respect to their peers using user and group profiles. Within AMC, users are assigned to roles that specify their job functions. We expect that users with the same role behave similarly and, thus, we use this feature to partition users in groups. To build behavior profiles for each user and group, first we need to select the event vectors that belong to a certain profile. In this respect, we define the concept of *profile constraint*. A profile is defined as a sequence of histograms.

DEFINITION 5 (PROFILE CONSTRAINT). *Given an event vector $\gamma \in \Gamma$, a profile constraint is defined as a function c that maps γ into $true$ or $false$. We define $\Gamma\,|_c$ as the subset of Γ containing event vectors for which c returns true, i.e.* $\Gamma|_c = \{\gamma \in \Gamma \mid c(\gamma) = true\}$.

DEFINITION 6 (PROFILE). *Given a set of event vectors $\Gamma \subseteq \Upsilon$, a profile constraint c, a feature space $F = \langle f_1, \ldots, f_n \rangle$ and the corresponding set of bin sequences $B = \langle \beta_{f_1}, \ldots, \beta_{f_n} \rangle$. The profile of c w.r.t. Γ is a sequence of histograms, i.e. $P_c(\Gamma\,|_c, F, B) = \langle \mu_{f_1}(\Gamma\,|_c, \beta_{f_1}), \ldots, \mu_{f_n}(\Gamma\,|_c, \beta_{f_n}) \rangle$.*

For example, by defining constraints $c_1 : UserID = 50009433$ and $c_2 : Role = Administratie\ (AZP)$ we can select event vectors belonging to user *50009433* and group *Administratie (AZP)* respectively. To make it easier for security officers to compare two profiles, we use bin fractions to graphically represent profiles.

DEFINITION 7 (BIN FRACTION). *Given a set of event vectors $\Gamma \subseteq \Upsilon$, a feature f and the corresponding bin sequence $\beta_f = \langle b_1, \ldots, b_n \rangle$, the bin fraction for a bin $b_i \in \beta_f$ is defined as $frac_{f,b_i} = \frac{freq_{f,b_i}(\Gamma)}{\sum_{j=1}^{|\beta_f|} freq_{f,b_j}(\Gamma)}$.*

Fig. 3 represents the profiles corresponding to c_1 (red) and c_2 (blue) built over the AMC log w.r.t. feature space $F_1 = \langle access\ type, time, division, date \rangle$. Feature *time* is partitioned into working ([8:00-18:00]) and non-working ([18:00-8:00]) hours.

3.3 Clustering and Subgroup Profiling

While users with a certain role are expected to act similarly, the experimental evidence shows that a predefined partitioning does not guarantee that users within a group actually behave similarly. For example, doctors in different departments might have different behaviors due to the different topology of patients they treat and to the different treatments they provide. Comparing users with heterogeneous behavior can provide misleading diagnostics. To limit this problem, we partition users with the same role into subgroups.

DEFINITION 8 (GROUP, SUBGROUP). *Let $\mathcal{U}$ be the set of all users. A group $G \subseteq \mathcal{U}$ is a subset of these users. A group sequence $GS = \langle G^1, \ldots, G^n \rangle$ partitions $\mathcal{U}$ into disjoint groups s.t. $\bigcup_{i=1}^{|GS|} G^i = \mathcal{U}$ and $\forall i \neq j, G^i \cap G^j = \emptyset$. Given a group G, a subgroup SG_G is a subset of users in G, i.e. $SG_G \subseteq G$. A subgroup sequence $SGS_G = \langle SG_G^1, \ldots, SG_G^n \rangle$ partitions G into disjoint subgroups s.t. $\bigcup_{i=1}^{|SGS_G|} SG_G^i = G$ and $\forall i \neq j, SG_G^i \cap SG_G^j = \emptyset$.*

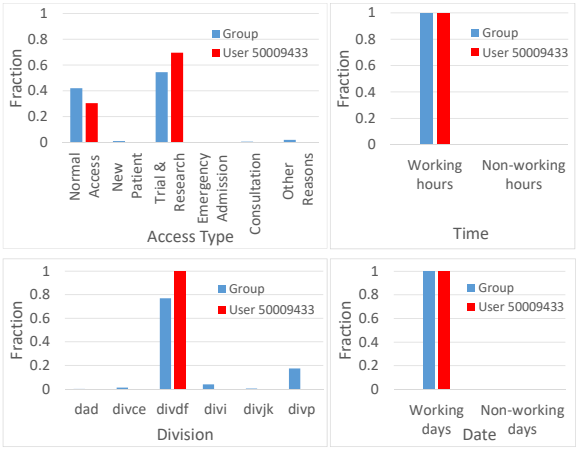

Figure 3: Profiles of user *50009433* and group *Administratie*

To partition users into subgroups, we use clustering. The problem of clustering has been widely studied and several clustering algorithms have been proposed (see [2] for a survey). Choosing an appropriate clustering algorithm and its parameters depends on the dataset and the purpose of analysis. Typically, several rounds of analysis are performed until the results with the desired properties are obtained.

In this work, we use X-means clustering [17] to find subgroups. This clustering method is based on k-means clustering [15], a popular clustering method that divides a group into k clusters. Every data point is assigned to the cluster where its distance to the mean of that cluster is the smallest, for a given distance metric. X-means clustering is different in the sense that parameter k does not need to be explicitly specified, merely an upper and a lower bound.

A problem we encountered in clustering is that of users with only few log entries. For example, some users work only few days in a month yielding a much lower number of log entries than users working at the hospital every day. The limited information available about these users can affect the quality of the subgroups constructed using clustering. To consider these users in the analysis while preserving the quality of clustering, we create a subgroup (hereafter called *Subgroup_s*) containing users with few events. The threshold used to determine the users to be assigned to *Subgroup_s* depends on the dataset and should be set by the analyst. For example, an analyst might set this threshold by taking into account the average number of actions performed by users.

In the AMC log, six users with role *Administratie (AZP)* executed less than 50 actions. We assigned these users to *Subgroup_s* and applied X-means clustering to partition the other users in the group. By doing so, we obtain two other subgroups, in addition to *Subgroup_s*. Fig. 4 shows the profiles of these subgroups. It is easy to observe that the profiles of these subgroups differ significantly from each other. Users in Subgroup_1 mostly accessed data from division *divp* and did not use the BTG procedure often, while users in Subgroup_2 mostly accessed data from division *divdf* and often

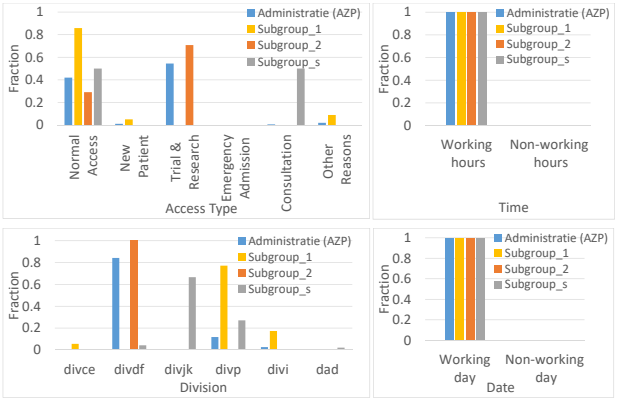

Figure 4: Subgroup profiles for users with role *Administratie*

used the BTG procedure to access data for *trial & research*. Users in Subgroup_s only accessed data from division *divjk* and half of these accesses were due to *consultation*. The similarity among users in these subgroups is that all of them accessed data during working hours and days. We can observe in Fig. 4 that the behavior of users in Subgroup_2 is similar to the group behavior. This is due to the fact that users in Subgroup_2 have a large number of accesses compared to users in other groups. Thus, the group profile is biased towards the behavior of these users while the behavior of other users with role *Administratie (AZP)* remain 'hidden' in the group profile.

3.4 Subgroup Profile Assessment

After building subgroup profiles, these profiles are analyzed to determine whether they match the behavior expected by users with the corresponding role. Ideally, this assessment is performed by domain experts. In this respect, histograms provide an easy way to graphically visualize and compare subgroup profiles.

DEFINITION 9 (ASSESSMENT FUNCTION). *Let $\mathcal{U}$ be the set of all users. Given a set of event vectors $\Gamma \subseteq \Upsilon$, a group sequence $GS \subseteq \mathcal{P}(\mathcal{U})^*$, and a subgroup $SG_G \in GS$, an assessment function λ maps SG_G into normal or anomalous, i.e. $\lambda : \mathcal{P}(\Upsilon) \times \mathcal{P}(\mathcal{U})^* \times \mathcal{P}(\mathcal{U}) \rightarrow \{normal, anomalous\}$*[1].

For example, consider the subgroup profiles in Fig. 4. Users with role *Administratie (AZP)* are not expected to provide consultations to other colleagues or perform research. In this case, an expert would mark the profile of Subgroup_1 as normal and the profiles of Subgroup_2 and Subgroup_s as anomalous.

However, analyzing profiles manually is time consuming and requires domain knowledge, which may not always be available. An alternative approach is to mark anomalous and normal profiles based on their structures. Following existing approaches [12, 18, 21], we assume that large subgroups are normal, while small subgroups are anomalous. Note that the number of users may vary by group to group; thus, the size of subgroups defined for them may also vary. To identify small subgroups, we have considered the relative size of subgroups, along the lines suggested in [12]. Given a group G, let $\Gamma\,|_c$ be the set of its event vectors and $SGS_G = \{SG_G^1, SG_G^2, \ldots, SG_G^k\}$ its subgroup sequence sorted by subgroup size, i.e. $|SG_G^1| \geq |SG_G^2| \geq \cdots \geq |SG_G^k|$. Given two parameters α and β, the boundary between large and small subgroups is b if one of the following formulae holds:

$$(|SG_G^1| + |SG_G^2| + ... + |SG_G^b|) \geq \alpha \times |\Gamma\,|_c\,| \quad (1a)$$

$$|SG_G^b|/|SG_G^{b+1}| \geq \beta \quad (1b)$$

where α determines the portion of users that should be taken as large clusters and β imposes the difference in size between large and small clusters. Then, the set of large subgroups is defined as $\{SG_G^1, \ldots, SG_G^b\}$ and the set of small subgroups is defined as $\{SG_G^{b+1}, \ldots, SG_G^k\}$.

It is worth mentioning that we always mark $Subgroup\_s$ as anomalous. This is because this subgroup is not identified through clustering and its users execute relatively few actions. After assessing the subgroups automatically, an expert can use histograms to review the results of automated subgroup assessment.

3.5 Anomaly Score Calculation

After assessing subgroups, we compute the anomaly score of each user by comparing his profile with the profile of his subgroup. Intuitively, the anomaly score measures to what extent a user's behavior differs from the one of his peers. This measure can be used to rank users for further investigation. In particular, an analyst with limited time can focus on the investigation of the most abnormal cases.

DEFINITION 10 (DISTANCE FUNCTION). *Let $\Gamma \subseteq \Upsilon$ be a set of event vectors, c_1 and c_2 two profile constraints, $F = \langle f_1, \ldots, f_n \rangle$ a feature space and B its corresponding set of bin sequences. A weight sequence $W = \langle w_{f_1}, \ldots, w_{f_n} \rangle$ is a sequence of real values s.t. $w_{f_i} \in [0, 1]$. Given two profiles $P_{c_1}(\Gamma\,|_{c_1}, F, B)$ and $P_{c_2}(\Gamma\,|_{c_2}, F, B)$ and a weight sequence, a distance function returns an anomaly score indicating the similarity between two profiles. We use κ to denote a distance function.*

The anomaly score for a user is determined by comparing his profiles with the profile of the subgroup the user belongs to. If the user acts in accordance with the subgroup profile, its anomaly score is low. Conversely, a user who acts radically differently from that subgroup will have a high anomaly score. Algorithm 1 represents how the anomaly score is computed. The algorithm keeps a list of users sorted with respect to their anomaly score. First, the algorithm initializes the list (line 1). Then, it iterates over all groups and assigns an anomaly score to each user (lines 2-18). In particular, for each group, the algorithm checks whether all subgroups are anomalous or not (lines 4-5). If at least one subgroup is not anomalous, the algorithm sets the value of $allAnomalousFlag$ to $false$ (line 5). Then, the algorithm iterates over the subgroups and compares user and subgroup profiles (lines 6-18). The subgroup that is being used for the comparison varies depending on the results of subgroup assessment. If a subgroup is marked as anomalous, its users are compared to the profiles of normal subgroups (lines 8-14). A user's anomaly score is the lowest anomaly score obtained from these comparisons. Otherwise, if the subgroup is normal, the profile of the user is compared with the subgroup that he belongs to (lines 16-18). Note that it is possible that all subgroups within a group are marked as anomalous. In this situation, similar to normal subgroups, the algorithm compares the users with their own subgroups. After assigning an anomaly score to

[1] We use $\mathcal{P}(X)$ to denote the power set of a set X.

Algorithm 1: Compute anomaly score

```
Input  : Set of event vectors Γ, Feature space F, Bin sequences B, Group sequence GS,
         Weight sequence W
Output : Anomaly list
1  list ← ∅;
2  foreach G ∈ GS do
3      allAnomalousFlag ← true;
4      foreach SG ∈ G do
5          if λ(Γ, GS, SG) = normal then allAnomalousFlag ← false ;
6      foreach SG ∈ G do
7          if λ(Γ, GS, SG) = anomalous and allAnomalousFlag = false then
8              foreach u ∈ SG do
9                  score ← ∞;
10                 foreach SG' ∈ G do
11                     if λ(Γ, GS, SG') = normal then
12                         score' ← κ(P_user=u(Γ|_user=u, F, B),
                               P_subgroup=SG'(Γ|_subgroup=SG', F, B), W);
13                         if score' ≤ score then score ← score' ;
14                 list.add(u, score);
15             else
16                 foreach u ∈ SG do
17                     score ← κ(P_user=u(Γ|_user=u, F, B),
                           P_subgroup=SG(Γ|_subgroup=SG, F, B), W);
18                 list.add(u, score);
19 return list;
```

each user, the algorithm returns the list of all users along with their anomaly scores (line 19). Note that, if a (sub)group has only one user, the analyst is required to determine whether the (sub)group and, thus, the user are anomalous. In this case, the user and (sub)group profiles coincide and the approach would return an anomaly score equal to 0.

It is worth mentioning that the features used for clustering can be different from the features used to compute the anomaly scores. The former features are selected to partition users with similar behavior into subgroups. On the other hand, the latter features determine the aspects relevant for the behavioral analysis. Different distance functions can be defined to compare two profiles. Next, we present two distance functions.

ΔF: Given a set of event vectors $\Gamma \subseteq \Upsilon$, a feature f and its corresponding bin sequence $B_{f_i} = \langle b_1, \ldots, b_n \rangle$, ΔF between the user profile obtained using profile constraint c_1 and the subgroup profile obtained using profile constraint c_2 with respect to feature f_i is computed as follows:

$$\Delta F_{f_i,c_1,c_2} = \sum_{i=1}^{|B_{f_i}|} |frac_{f_i,b_i}(\Gamma\,|_{c_1}) - frac_{f_i,b_i}(\Gamma\,|_{c_2})| \qquad (2)$$

After computing the distances between two profiles with respect to each feature, we need to combine them. To combine these values, we can weight features based on their importance. We define ΔF between these two profiles with respect to feature space $F = \langle f_1, \ldots, f_n \rangle$ and its corresponding weight sequence $\langle w_{f_1}, \ldots, w_{f_n} \rangle$ as the weighted sum of ΔF computed for defined feature, i.e. $\Delta F_{F,c_1,c_2} = \sum_{i=1}^{|F|} w_{f_i} \times \Delta F_{f_i,c_1,c_2}$.

As an example, consider the profiles of Subgroup_1 in Fig. 4 and user 50009433 in Fig. 3. If we use ΔF to compare these two profiles w.r.t. features *access type*, *time*, *division* and *date*, we obtain 1.39, 0, 2 and 0 respectively. If the weights assigned to features are set to 1, then the value of ΔF for these profiles is equal to 3.39.

This distance function is straightforward, but has the problem that it only considers the fraction of bins and neglects their frequencies when it compares two profiles. For example, consider the profile of user 50009433 in Fig. 3 and the profile of Subgroup_1 in Fig. 4. Suppose that a hypothetical user, hereafter called user 100, has the same behavior of user 50009433 in terms of distribution but has twice as many accesses as user 50009433. If we compare the profiles of these users with Subgroup_1 using distance function ΔF, the outcome is the same. One may argue that since user 100 retrieved more information for *trial & research* than user 50009433 (while Subgroup_1 never accessed data for this purpose), his behavior may be more interesting to investigate; thus, the distance function should assign a higher value to it. Next, we introduce another distance function that takes into account the frequency of bins.

χ^2 Score: A distance function that takes into account the number of actions performed by users is the χ^2 test statistic [9], henceforth the χ^2 score. Given a set of event vectors $\Gamma \subseteq \Upsilon$, a feature f_i and its corresponding bin sequence $B_{f_i} = \langle b_1, \ldots, b_n \rangle$, the χ^2 score between a user's profile obtained using profile constraint c_1 and a subgroup profile obtained using profile constraint c_2 w.r.t. feature f_i is computed as follows:

$$\chi^2_{f_i,c_1,c_2} = \sum_{i=1}^{|B_{f_i}|} \frac{(freq_{f_i,b_i}(\Gamma\,|_{c_1}) - frac_{f_i,b_i}(\Gamma\,|_{c_2}) \times |\Gamma\,|_{c_1}\,|)^2}{frac_{f_i,b_i}(\Gamma\,|_{c_2}) \times |\Gamma\,|_{c_1}\,| + \epsilon} \qquad (3)$$

Note that the frequency of a bin might be zero in a histogram. To avoid division by zero, we add a small number $\epsilon \in \mathbb{R}^+$ to the denominator. This number, however, may significantly affect the computed χ^2 scores. The choice for the value of ϵ depends on to what extent a non-zero bin in a user profile should be penalized when the corresponding bin in a subgroup profile is zero. For example, if the chosen ϵ is too small, the χ^2 scores computed for this type of user profiles become much higher than other user profiles.

The χ^2 scores between two profiles w.r.t. feature space $F = \langle f_1, \ldots, f_n \rangle$ and its corresponding weight sequence $W = \langle w_{f_1}, \ldots, w_{f_n} \rangle$ is the weighted sum of the χ^2 scores w.r.t. all features, i.e. $\chi^2_{F,c_1,c_2} = \sum_{i=1}^{|F|} w_{f_i} \times \chi^2_{f_i,c_1,c_2}$.

As an example, consider the profiles of Subgroup_1 in Fig. 4, user 50009433 in Fig. 3 and user 100 (recall that user 100 has the same behavior of user 50009433 in terms of distribution but performed twice his actions). By setting $\epsilon = 0.1$ and $W = \langle 1, 1, 1, 1 \rangle$, we obtain the χ^2 score 50055732.34 for user 50009433 and 200217425.2 for user 100. Thus, the χ^2 score meets our desideratum of discriminating user behavior with respect to the number of actions users performed.

4 UNDERSTANDING THE LOG

Our analysis of the use of the BTG procedure at AMC relies on a log recorded over one month (May 2015). The log comprises 1,059,404 events recording the accesses of 4,603 users to the medical records of 87,666 patients. According to the log, there are 50 roles and 21 divisions within the hospital. In total, users invoked the BTG procedure 53,567 times (5.2% of all accesses). Information of 23% of the patients were accessed using this procedure and 38% of the users invoked it at least once to access patient information.

To understand how the BTG procedure is used at AMC, we applied various data analysis techniques to the log. Fig. 5 visualizes the use of the BTG procedure over time using a dotted chart. In this chart, the horizontal axis shows time and the vertical axis shows patients. A dot in the chart indicates that patient information was accessed using the BTG procedure. The color of dots represents the reason provided by users at request time. We can observe many more data was accessed during working days compared to non-working

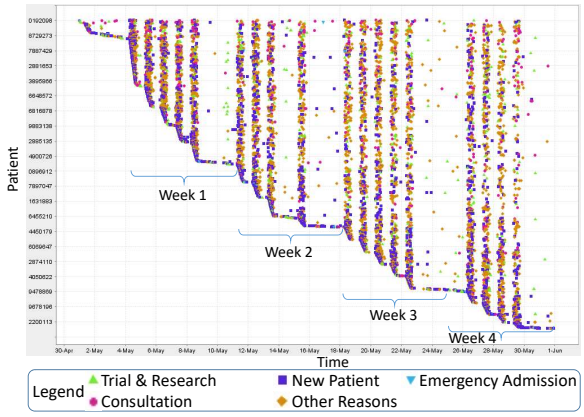

Figure 5: Invocations of the BTG procedure over time.

days (similar observation is obtained for normal accesses). It is worth mentioning that May 14th and 25th are national holidays in the Netherlands. That is why few events were recorded on those days. During non-working days, the data of new arrival patients were mostly accessed whereas very few accesses were performed on the data of patients that were already admitted to the hospital.

A similar pattern can also be observed by comparing accesses executed during working and non-working hours of working days. This observation suggests that users exhibit a different behavior based on the days and hours that they work at the hospital. Moreover, by performing a similar analysis on users with the same role working in different divisions, we observed that their behavior differs significantly (these dotted charts are not reported here due to the lack of space).

These findings were used in our experiments (Section 5) for feature selection and binning. In particular, in addition to access type, we also consider date, time, and division as features. Moreover, our analysis shows that, although there is a clear difference between working and non-working hours/days, the difference is negligible during working hours/days (and non-working hours/days). Thus, we binned the feature values of the time and date features accordingly.

To analyze how a user accesses patient data, we extract the sequences of accesses that each user performed on each patient's data. Hereafter, these sequences are called *traces*. We used Disco (`https://fluxicon.com/disco/`) to discover a model for these traces. Disco is a process mining tool that can mine a fuzzy model from a given log. Fig. 6 shows a simplified model of 40% of the paths mined by Disco. The thickness of paths shows the time interval between the execution of two activities. Note that every time that a user invokes the BTG procedure, two events are recorded. The *access type* of the first event is set to *request for using BTG procedure* and the one of the second event is set to the selected access reason. The model shows that BTG accesses can be followed by normal accesses. Fig. 6a represents an example of these traces, where first the user chose inter-colleague consultation as a reason to access data and later he accessed data normally. After analyzing the time intervals between accesses and discussing our findings with experts at AMC, we realized that, after users invoke the BTG procedure to access data, they can access the data again without reusing this procedure within 24 hours.

Moreover, the model in Fig. 6 shows that a user might use the BTG procedure multiple times to access information of the same patient. Fig. 6b represents an example of these accesses. Three explanations were provided about this finding to us. First, users may access patient information for different reasons (e.g., research or providing treatment). Although users can potentially access data normally, they are supposed to request the data using the BTG procedure and specify a new reason for the access if they want to use the data for other purposes. The second explanation is that the time interval between two accesses is more than 24 hours. The third explanation is that patient records consist of different parts with different sensitivities. Users have to reuse the BTG procedure to access a different portion of a patient record. For example, a user may first use the BTG procedure to view demographic information of a patient. Then, the BTG procedure should be used again by the user to access the patient's medical history. Note that highly sensitive information such as psychiatric information cannot be accessed using the BTG procedure.

Another behavior that can be observed in the model of Fig. 6 is that some accesses were first denied but later granted. Fig. 6c represents an example of these traces. By discussing this with AMC, it turned out that users can access the record of only one patient at a time. In these cases, access was denied because the user tried to access the information of a patient while having the window showing information of another patient open.

Based on the analysis of the log and a discussion of our findings with AMC, we were able to refine $Policy_3$ and expand the policies initially elicited (Section 2). Medical records are divided into different parts. In order to access each part of a medical record of a patient registered at a different department, a user has to use the BTG procedure. This is independent of whether the user used this procedure to access other parts of the medical record ($Policy_3'$). After using the BTG procedure to access a certain part of a patient record, the user can access it again without invoking the BTG procedure within 24 hours after the first access ($Policy_4$). Highly sensitive patient data cannot be accessed using the BTG procedure ($Policy_5$). Users can only view information of one patient at a time ($Policy_6$).

5 RESULTS AND DISCUSSION

We implemented the approach illustrated in Fig. 2 as a plug-in of the RapidMiner framework (`https://rapidminer.com/`). The plug-in takes as input a log and computes the anomaly score for each user. The output of the plug-in consists of user, subgroup and group profiles, which can be used by other tools for further analysis. A screenshot of our tool is shown in Fig. 7. In the figure, the top left table shows the list of users ranked according to their anomaly score. Security analysts can focus on the analysis of a certain group, a certain subgroup, or all users. The details of the subgroups of the selected group are shown in the top right table. The charts in the second and third rows show the profiles of the user alongside the one of its group and subgroup. The chart at the bottom shows the number of data accesses on different days. We applied our approach to the AMC log and discussed our findings with domain experts at the hospital.

Based on the insights obtained from the data analysis in Section 4, we defined feature space $F = \langle access\ type, time, division, date \rangle$. We removed from the log all events in which *access type* is set to *access denied* or *request for using BTG procedure*. The bin sequence for feature *time* is defined as $\langle working\ hours, non\text{-}working\ hours \rangle$

	Activity	Date	Time
1	Request for using BTG procedure	04.05.2015	11:18:43
2	Intercolleague consultation	04.05.2015	11:18:50
3	Access without using BTG procedure	04.05.2015	15:54:39

(a) Normal data access after using the BTG procedure

	Activity	Date	Time
1	Request for using BTG procedure	01.05.2015	16:06:06
2	New patient	01.05.2015	16:06:11
3	Access without using BTG procedure	01.05.2015	16:12:44
4	Request for using BTG procedure	01.05.2015	16:29:17
5	New patient	01.05.2015	16:29:19
6	Access without using BTG procedure	01.05.2015	16:30:43

(b) Reusing the BTG procedure

	Activity	Date	Time
1	Access denied	08.05.2015	09:48:50
2	Access without using BTG procedure	08.05.2015	10:14:08
3	Access without using BTG procedure	08.05.2015	15:31:29

(c) First access denied and later granted

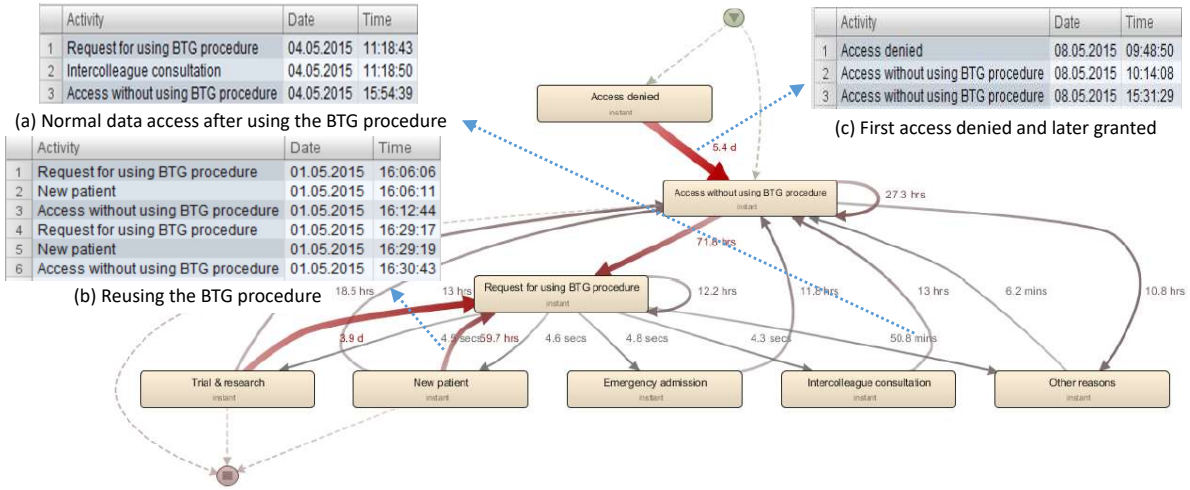

Figure 6: Fuzzy model mined by Disco with high abstraction level in paths. Each trace contains all the events related to a patient and a user. The label of arcs represents average time between the occurrence of events.

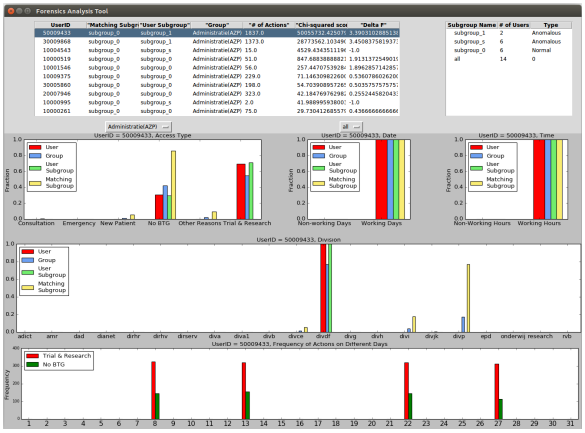

Figure 7: Screenshot of the tool

and the bin sequence for feature *date* is defined as ⟨*working days*, *non-working days*⟩. We grouped users based on their roles and partitioned them into subgroups using X-means clustering. For clustering, the minimum and maximum number of clusters were set to 2 and 10, respectively. For each group, users with less than 50 actions were assigned to $Subgroup\_s$. This subgroup was always marked as anomalous. To assess other subgroups, we used the approach presented in Section 3.4 (see Eqs. 1a and 1b). We set parameter α to 0.9 under the assumption that most users behave normally; β was set to 3 to ensure that the size of any large subgroup is at least three times the size of small subgroups. The two distance functions presented in Section 3.5 were used to compute anomaly scores. To compute the χ^2 score, we set $\epsilon = 0.1$. We considered two weight sequences and performed two experiments. In the first experiment, all weights were set to 1. In the second, we adjusted the weights based on the importance of features.

5.1 Quantitative Analysis

In this section, we present our results. For both experiments, we partitioned users with the same role into subgroups. In total, we found 116 subgroups. Out of these subgroups, 82 subgroups were found by the X-means algorithm and 34 subgroups are subgroups containing users with less than 50 actions ($Subgroup\_s$). Our approach marked 69 subgroups as normal and 47 subgroups as anomalous. It is worth noting that, while the settings of the X-means algorithm allowed from 2 to 10 subgroups, every group was partitioned in at most 4 subgroups (ignoring $Subgroup\_s$). From these results we conclude that the choice of clustering parameters is appropriate.

In the first experiment, we assigned the same weight to all features, i.e. $W = \langle 1, 1, 1, 1 \rangle$. Fig. 8 shows ΔF computed for the users. Note that since we use four features in this experiment, the maximum score that this distance function assigns to a user is 8. The results show that most of the users with the highest ΔF executed few actions. Recall that this distance function considers the bin fractions when comparing two histograms. If few events are recorded for a user, the corresponding feature values for these events may fall only in few bins. Thus, the fraction of few bins can be very high and the rest 0, which might be very different from the fractions computed for the matching subgroup. Fig. 9 shows the χ^2 score assigned to users. In contrast to ΔF, the χ^2 score considers the bin frequencies when comparing two histograms. That is why most of the users with the highest χ^2 score are not members of $Subgroup\_s$.

Recall that the χ^2 score is the weighted sum of the χ^2 scores w.r.t. each feature. Different features can have different degrees of influence on the total χ^2 score. Given a feature $f_i \in F$, we use r_f to denote the ratio of the χ^2 score of a feature ($\chi^2_{f_i,c_1,c_2}$) to the total χ^2 score (χ^2_{F,c_1,c_2}), i.e. $r_f = \frac{\chi^2_{f_i,c_1,c_2}}{\chi^2_{F,c_1,c_2}}$. An analysis of the results obtained in the first experiment shows that this ratio for features *access type*, *time*, *division* and *date* is 0.273, 0.002, 0.723 and 0.002, respectively. These ratios indicate that the computed χ^2 scores primarily depend on the *division*. This can be explained by the nature of this feature. Most users only executed actions from one or two divisions, while there are many divisions in the hospital. For a large group, such as *BA-Behandelend Arts* or *Verpleging (VPK)*, these accesses were executed from different divisions. Moreover, due to

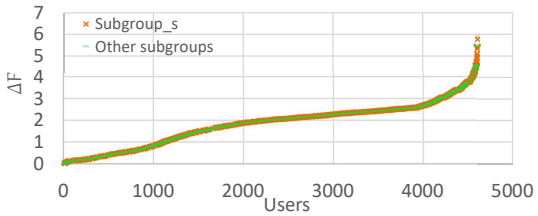

Figure 8: ΔF assigned to users.

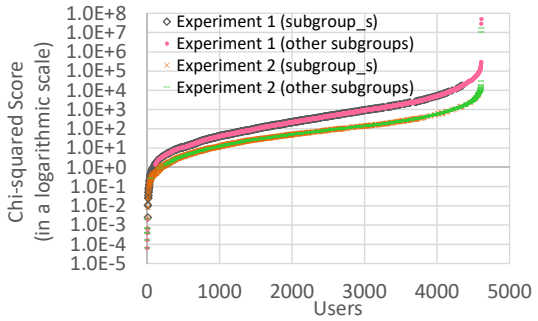

Figure 9: χ^2 score assigned to users.

the large amount of possible outcomes, some of the bin fractions of this feature are likely to be zero. As this number is used in the denominator of Eq. 3, this can lead to very high χ^2 scores.

As experts suggested, we balanced the effect of feature *division* on the total χ^2 score. In particular, we set the ratio of the χ^2 score for this feature over the total χ^2 score to 0.1. To this end, we used weight sequence $W = \langle 1, 1, 0.04, 1 \rangle$ and performed the experiment again. Fig. 9 shows the new χ^2 score assigned to users (bottom line). The average χ^2 score is 6,511, which is almost half of the average χ^2 score obtained from the previous experiment.

5.2 Qualitative Analysis

We discussed our findings with six domain experts with security backgrounds at AMC and evaluated our results based on their feedback. Here, we report some of these findings.

Administratie (AZP) and GAZP-Gedel. Adm. Zorgproces: Users with these roles are responsible for the admission of patients to the hospital. These users are not supposed to access data for purposes such as consultation or research. Nonetheless, among users with role *Administratie (AZP)*, two users accessed data for research several times. These users also accessed data from division *divdf*, while other users with this role never accessed data from this division. Similarly, a user with role *GAZP-Gedel. Adm. Zorgproces* accessed data 299 times for research, which is 53% of the user's total accesses, whereas most users in this group never used the BTG procedure. According to the experts at AMC, one reason for which research could have often been selected by users is that this is the default option. Thus, some users might choose that option instead of selecting the actual reason for which they use the BTG procedure. Another explanation provided by the experts is that the wrong role might have been assigned to those users. This can explain why they behave very differently from other users with the same role. In any case, our analysis was deemed useful to correct procedural mistakes.

CO-co assistent: This group comprises medical students. These users are typically assigned to various projects in different divisions as part of their training. We observed that some users in this group often used the BTG procedure to access patient data. For example, a user accessed data 634 times (43% of the user's accesses) by selecting *new patient* as the access reason and another user accessed data 269 times (81% of the user's accesses) for research. These accesses were concentrated in a few days. By discussing this observation with domain experts at AMC, it turned out that, at the beginning of a new research project, users may collect a large amount of patient information and store it locally.[2] This local copy of data can then be accessed without invoking the BTG procedure. Moreover, when assigned to another division, medical students might still need to access patient data from the previous division to perform more analysis and complete their previous project. That is why some of them used the BTG procedure to access patient data, although this behavior should not be allowed according to the experts at AMC.

BA-behandelend arts: These users are doctors. The log shows that one of these users performed many more data accesses compared to other doctors. Specifically, this user accessed patient data 15,798 times, while other users on average accessed patient data 304 times. These accesses were performed every day and at any time. After discussing this finding with the experts at AMC, it turns out that software applications are used to check patient data periodically.

Lab1: Users with this role are responsible for performing lab tests. Among these users, a user accessed data from four different divisions. No other user in this group accessed data from three of these divisions. According to experts at AMC, a possible explanation for this finding is that this user has shared his/her credentials with users working in different divisions.

All six experts found it easy to understand and compare histograms representing users, subgroups and groups' profiles. They also acknowledged that our findings can be useful for the hospital for several reasons. First, they can assist in the identification of data misuses. Moreover, they can help fixing possible errors in the system configuration. For example, our approach allows identifying possible mistakes in assigning roles to users. Finally, they can help correcting user behavior if users do not use the system as expected. For example, the results of our analysis showed that some users select the default option when they use the BTG procedure.

6 RELATED WORK

Current practices have shown the importance of flexible security mechanisms in hospitals and this need has attracted significant attention in the research community. A research stream aims to extend existing access control models with the BTG procedure [3, 6, 11, 20]. Ferreira et al. [11] integrate BTG features within Role-Based Access Control. Brucker et al. [6] extend SecureUML to support the BTG procedure and propose a security architecture supporting this procedure. Rissanen et al. [3] present a discretionary overriding mechanism in XACML. Marinovic et al. [16] propose a BTG policy language called *Rumpole* to specify how override requests should be handled. Schefer-Wenzl and Strembeck [20] present a BTG extension for business process models. These approaches, however, only

[2]Note that, in this case, the collected patient records should be anonymized. However, it is not possible to verify whether this was actually the case using the available log.

focus on how to incorporate the BTG procedure into an IT system and do not study how this procedure is actually used.

Another research stream has focused on methods to analyze and control the use of the BTG procedure [1, 5]. For instance, Azkia et al. [5] show how events recorded in the IHE-ATNA log format can be analyzed to identify possible violations of Organization-Based Access Control policies. Adriansyah et al. [1] propose a framework based on the notion of alignments to control and limit what users can do when the BTG procedure is used. The framework allows users to deviate from the specification only if the severity of deviations can be tolerated. Compared to our work, these approaches require predefined policies or process models to analyze user behavior.

Our work has similarities with proposals in the field of anomaly detection. This field has been widely studied in the literature and applied to different domains such as credit card fraud detection, network intrusion detection and critical system fault detection [7, 14]. Anomaly detection techniques can be divided into supervised, semi-supervised and unsupervised [7]. Supervised and semi-supervised approaches assume that training data contains labeled events. In contrast, in unsupervised approaches, no labels are needed. As it is very difficult to obtain labeled events for data accesses in hospitals, supervised and semi-supervised approaches cannot be applied in our context. Two unsupervised techniques closely related to our work are clustering and histogram-based methods. Clustering methods [12, 13] group similar instances according to a distance function and mark small or sparse clusters as anomalous. These techniques, similar to other anomaly detection techniques [7, 14], can have a high false positive rate. Thus, when an alert is raised, it has to be investigated to determine whether it corresponds to an actual attack. However, these techniques often provide very little or no support for alert handling, resulting in high operational costs. This issue is addressed by histogram-based approaches. For instance, Costante et al. [8] propose a white-box anomaly detection approach based histograms. This approach learns profiles representing normal behavior from past transactions based on predefined features and compares new transactions against these profiles to identify anomalies. However, as discussed in Section 3, the construction of profiles based on predefined features can provide misleading diagnostics. In this work, we have combined the use of clustering and histograms. In particular, clustering is used to group users exhibiting similar behavior, leading to the construction of more accurate profiles and, thus, enhancing detection capabilities. On the other hand, histograms are used to facilitate the understanding and analysis of suspicious behavior. Moreover, differently from anomaly detection techniques that analyze each transaction independently, our approach supports the analysis of the overall behavior of a user and, thus, allows the detection of attacks that otherwise can remain undetected.

7 CONCLUSION

In this paper, we have presented an approach for behavior analysis and applied it to study the use of the Break-The-Glass (BTG) procedure in a large Dutch hospital. Our approach constructs behavioral profiles of users and compares these profiles with users' expected behavior represented by the profile of users with the same job function. We use off-the-shelf clustering algorithms to group users that exhibit similar behavior, leading to the construction of more accurate profiles and, thus, enhancing detection capabilities. To validate the applicability of our approach in real-life settings, we analyzed a log collected from AMC. A discussion of our findings with experts at AMC showed that our approach provides meaningful insights on user behavior. Experts also found histogram-based profiles easy to understand and useful for the analysis of abnormal behavior. As future work, we intend to evaluate the proposed approach using logs collected from different hospitals over a longer period of time. Moreover, we plan to investigate its application for real-time analysis. This would allow earlier detection and response to suspicious behavior.

Acknowledgments. This work has been partially funded by the NWO CyberSecurity programme under the PriCE project.

REFERENCES

[1] Arya Adriansyah, Boudewijn F van Dongen, and Nicola Zannone. 2013. Controlling Break-the-Glass through Alignment. In *Proceedings of International Conference on Social Computing*. IEEE, 606–611.
[2] Charu Aggarwal and Chandan Reddy. 2013. *Data clustering: algorithms and applications*. CRC press.
[3] Ja'far Alqatawna, Erik Rissanen, and Babak Sadighi. 2007. Overriding of Access Control in XACML. In *Proceedings of International Workshop on Policies for Distributed Systems and Networks*. IEEE, 87–95.
[4] Ajit Appari and M Eric Johnson. 2010. Information security and privacy in healthcare: current state of research. *IJIEM* 6, 4 (2010), 279–314.
[5] Hanieh Azkia, Nora Cuppens-Boulahia, Frédéric Cuppens, Gouenou Coatrieux, and Said Oulmakhzoune. 2015. Deployment of a posteriori access control using IHE ATNA. *International Journal of Information Security* 14, 5 (2015), 471–483.
[6] Achim D Brucker and Helmut Petritsch. 2009. Extending access control models with break-glass. In *Proc. of SACMAT*. ACM, 197–206.
[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *Comput. Surveys* 41, 3 (2009), 15.
[8] Elisa Costante, Jerry den Hartog, Milan Petkovic, Sandro Etalle, and Mykola Pechenizkiy. 2017. A white-box anomaly-based framework for database leakage detection. *J. Inf. Sec. Appl.* 32 (2017), 27–46.
[9] Wayne W Daniel and Chad Lee Cross. 1995. *Biostatistics: a foundation for analysis in the health sciences*. Wiley New York.
[10] European Union. 2016. General Data Protection Regulation. `http://data.europa.eu/eli/reg/2016/679/oj`. (2016). Accessed: 2017-10-15.
[11] Anna Ferreira, Ricardo Cruz-Correia, Luis Antunes, Pedro Farinha, E Oliveira-Palhares, David Chadwick, and Altamiro Costa-Pereira. 2006. How to break access control in a controlled manner. In *Proc. of Computer-Based Medical Systems*. IEEE, 847–854.
[12] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters* 24, 9 (2003), 1641–1650.
[13] Mon-Fong Jiang, Shian-Shyong Tseng, and Chih-Ming Su. 2001. Two-phase clustering process for outliers detection. *Pattern Recognition Letters* 22, 6 (2001), 691–700.
[14] Hung-Jen Liao, Chun-Hung Lin, Ying Lin, and Kuang Tung. 2013. Intrusion detection system: A comprehensive review. *J. of Network and Computer Applications* 36, 1 (2013), 16–24.
[15] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. 281–297.
[16] Srdjan Marinovic, Naranker Dulay, and Morris Sloman. 2014. Rumpole: An introspective break-glass access control language. *TISSEC* 17, 1 (2014), 2.
[17] Dan Pelleg and Andrew W Moore. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings of International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 727–734.
[18] Ana M Pires and Carla Santos-Pereira. 2005. Using Clustering and Robust Estimators to Detect Outliers in Multivariate Data. In *Proc. of Int. Conf. on Robust Statistics*.
[19] Ponemon Institute 2015. Cost of data breach study: global analysis. (2015). `https://www.ponemon.org/library/sixth-annual-benchmark-study-on-privacy-security-of-healthcare-data-1`.
[20] Sigrid Schefer-Wenzl and Mark Strembeck. 2014. Model-driven specification and enforcement of RBAC break-glass policies for process-aware information systems. *Information and Software Technology* 56, 10 (2014), 1289–1308.
[21] Karlton Sequeira and Mohammed Zaki. 2002. ADMIT: anomaly-based data mining for intrusions. In *Proc. of KDD*. ACM, 386–395.