Role Mining with Missing Values

Sokratis Vavilis*, Alexandru Ionut Egner*, Milan Petković*† and Nicola Zannone*
*Eindhoven University of Technology
Email: {s.vavilis, a.i.egner, n.zannone}@tue.nl
†Philips Research Europe
Email: milan.petkovic@philips.com

***Abstract*—Over the years several organizations are migrating to Role-Based Access Control (RBAC) as a practical solution to regulate access to sensitive information. Role mining has been proposed to automatically extract RBAC policies from the current set of permissions assigned to users. Existing role mining approaches usually require that this set of permissions is retrievable and complete. Such an assumption, however, cannot be met in practice as permissions can be hard-coded in the applications or distributed over several subsystems. In those cases, permissions can be obtained from activity logs recording the actions performed by users. This, however, can provide an incomplete representation of the permissions within the system. Thus, existing role mining solutions are not directly applicable. In this work, we study the problem of role mining with incomplete knowledge. In particular, we investigate approaches for two instances of the role mining problem with missing values. Moreover, we study metrics to properly evaluate the obtained RBAC policies. We validate the investigated approaches using both synthetic and real data.**

***Index Terms*—Role Mining, RBAC, incomplete knowledge, metrics.**

I. Introduction

Over the years, organizations are challenged to protect an increasing amount of sensitive information. To this end, they are seeking secure and practical solutions to regulate the access to those information. Role-Based Access Control (RBAC) has proven to be a suitable candidate to manage authorizations due to its flexibility and ability to capture an organization's structure and job functions. According to NIST, RBAC is part of several security standards for various fields, such as Industrial Control Systems, Military, Biometrics and Healthcare.[1] For instance, the adoption of RBAC is proposed by many standardization initiatives and legislations, such as HIPAA[2] and HL7[3], to facilitate the management of sensitive medical information. The economic benefits led by the adoption of RBAC has been estimated in $1.1 billion [1].

Migrating to an RBAC system, however, is not a trivial task as the correct definition of roles is crucial for the successful deployment of such a system [2]. Two main approaches have been proposed to define roles along with the permissions that should be assigned to roles: role engineering and role mining. Role engineering aims to derive roles by manually analyzing organization policies and business processes. Unfortunately, role engineering cannot be automated as significant human intervention is required to define roles. This issue is addressed by role mining. Role mining typically relies on data mining techniques to discover roles from the analysis of the current set of permissions assigned to users.

Role mining approaches usually assume that the set of permissions assigned to users is directly accessible and complete. However, this assumption is difficult to meet in practice. For instance, the specification of users' permissions might be distributed among different subsystems of the organization. Moreover, permissions can be hard-coded into enterprise applications and legacy systems. Thus, existing role mining approaches cannot be directly applied to extract the RBAC policy. The set of permissions assigned to users can be obtained by analyzing the actions they performed within the system. In particular, user actions, such as access to organization resources, are usually captured by IT systems in activity logs. These logs, however, may provide only an incomplete representation of the permissions within the system, as it is unlikely that users would have exercised all their permissions.

In this work, we address the problem of role mining with incomplete knowledge. In particular, we analyze two instances of the role mining problem (RMP) with missing values, namely minimal noise (MinNoise) RMP and multiple factor optimization (MFO) RMP. The MinNoise RMP [3] aims to discover roles by minimizing the approximation error with respect to the set of permissions assigned to users, given an estimation of the number of roles. On the other hand, the MFO RMP is a problem inspired by approaches in the matrix factorization domain [4]. This RMP provides the best solution with respect to the description length of several aspects such as the number of roles and approximation error. The description length, however, does not fit the nature of role mining as it is expressed in terms of number of bits used to represent the solution. To this end, we present an evaluation metric tailored to the needs of role mining. To facilitate the migration to RBAC we investigate practical approaches for the MinNoise and MFO RMPs with missing values. We have validated the proposed approach using both synthetic and real data.

The remainder of the paper is organized as follows. The next section presents preliminaries on RBAC and role mining. Section III motivates the need of approaches for role mining with incomplete knowledge using an example in the healthcare domain. Section IV provides a formal representation of two role mining problems with missing values. Section V discusses approaches to solve these RMPs and presents a metric for

[1]http://csrc.nist.gov/groups/SNS/rbac/standards.html
[2]hhttp://www.hhs.gov/hipaa/for-professionals/security/laws-regulations
[3]http://www.hl7.org/implement/standards/product_brief.cfm?product_id=72

the evaluation of the elicited RBAC policies. An evaluation of the studied approaches is presented in Section VI. Finally, Section VII discusses related work and Section VIII concludes the paper by providing directions for future work.

II. Preliminaries

A. Basic definitions and Notation

This section presents preliminaries on RBAC along with the notation used in the paper.

Definition 1 (RBAC)**.**

- *U, R, OPS, and OBJ are the set of users, roles, operations and objects, respectively.*
- $PRMS \subseteq \{(op, obj) | op \in OPS \wedge obj \in OBJ\}$ *is the set of permissions.*
- $UA \subseteq U \times R$ *is the user-role assignments.*
- $PA \subseteq R \times PRMS$ *is the role-permission assignments.*
- $UPA \subseteq U \times PRMS$ *is the user-permission assignments.*

Definition 2 (RBAC State)**.** *An RBAC state is a tuple* (R, UA, PA).

Assignments *UA*, *PA* and *UPA* can be represented in the form of Boolean matrices. Given m users, n permissions and k roles, a user-role assignment *UA* can be represented as an $m \times k$ Boolean matrix, where $UA_{ij} = 1$ indicates that user i is assigned to role j and $UA_{ij} = 0$ indicates that user i is not assigned to role j. Similarly, *PA* can be represented as a $k \times n$ Boolean matrix and *UPA* as an $m \times n$ Boolean matrix. Hereafter, we denote $M(UA)$, $M(PA)$ and $M(UPA)$ the Boolean matrix representation of *UA*, *PA* and *UPA* respectively. Next, we present the Boolean Matrix arithmetic used in role mining.

Definition 3 (Boolean Matrix Multiplication)**.** *A Boolean matrix multiplication between an* $m \times k$ *Boolean matrix A and a* $k \times n$ *Boolean matrix B is* $A \otimes B = C$*, where C is an* $m \times n$ *Boolean matrix such that* $c_{ij} = \bigvee_{l=1}^{k} (a_{il} \wedge b_{lj})$

Definition 4 (L_1 Norm)**.** *The* L_1 *norm of an* $m \times n$ *matrix A is* $||A||_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|$.

The L_1 norm of the difference between two Boolean matrices indicates the similarity of the matrices. In particular, given two Boolean matrices A and B, the Boolean matrix $C = A - B$ represent the difference of matrices A and B. A cell C_{ij} is equal to 0 only when $A_{ij} = B_{ij}$. Thus, the L_1 norm of matrix C represents the number of cells in C for which matrices A and B have a different value.

The L_1 norm makes it possible to measure the consistency of an RBAC state [3]:

Definition 5 (δ-Consistency)**.** *A user-role assignment UA, a role-permission assignment PA and a user-permission assignment UPA are* δ*-consistent if and only if:*

$$||M(UA) \otimes M(PA) - M(UPA)||_1 \leq \delta \tag{1}$$

B. Inventory of RMP

This section presents an overview of the main role mining problems proposed in the literature [3], [5].

Definition 6 (Role Mining Problem [3])**.** *Given a set of users U, a set of permissions PRMS and a user-permission assignment UPA, find a set of roles R, a user-role assignment UA and a role-permission assignment PA 0-consistent with UPA and minimizing the number of roles k.*

Intuitively, the Role Mining Problem (RMP) is a minimization problem that aims to find a minimal set of roles R, a user-role assignment *UA* and a role-permission assignment *PA* from a user-permission assignment *UPA*. In matrix notation it can be formulated as:

$$M(UPA) = M(UA) \otimes M(PA) \tag{2}$$

Definition 7 (δ-Approx RMP [3])**.** *Given a set of users U, a set of permissions PRMS, a user-permission assignment UPA and a threshold* δ*, find a set of roles R, a user-role assignment UA and a role-permission assignment PA* δ*-consistent with UPA and minimizing the number of roles k.*

The δ-Approx RMP aims to find the minimal set of roles R, a user-role assignment *UA* and a role-permission assignment *PA* from a user-permission assignment *UPA*, while maintaining the approximation error below δ (see Definition 5). It is worth noting that the basic RMP is an instance of the δ-Approx RMP where $\delta = 0$.

Definition 8 (Minimal Noise RMP [3])**.** *Given a set of users U, a set of permissions PRMS, a user-permission assignment UPA and a number of roles k, find a set of k roles R, a user-role assignment UA and a role-permission assignment PA, minimizing*

$$||M(UA) \otimes M(PA) - M(UPA)||_1 \tag{3}$$

The Minimal Noise (MinNoise) RMP is a minimization problem that aims to determine a set of k roles R, a user-role assignment *UA* and a role-permission assignment *PA* from a user-permission assignment *UPA*, which minimizes the approximation error. Intuitively, MinNoise RMP finds the best solution with respect to the approximation error, for a particular number of roles.

Definition 9 (Minimal Edge RMP [5])**.** *Given a set of users U, a set of permissions PRMS and a user-permission assignment UPA, find a set of roles R, a user-role assignment UA and a role-permission assignment PA 0-consistent with UPA and minimizing*

$$||M(UA)||_1 + ||M(PA)||_1 \tag{4}$$

Intuitively, the minimization of $||M(UA)||_1 + ||M(PA)||_1$ provides the set of roles that requires the minimum number of assignments, thus reducing the administrative burden of managing the roles.

It is worth noting that most of the role mining problems aim to find a set of roles by optimizing only one particular

aspect. However, the role mining problem can be extended to mine roles optimizing several different aspects. Although this issue is not directly addressed in the role mining literature, an interesting approach is proposed in the Boolean matrix factorization field. The Boolean matrix factorization problem (BMF) can be seen as the counterpart of MinNoise RMP in the data mining domain [3], [6]. To provide higher quality factorization, a recent work on BMF [4] has proposed the minimization with respect to a more complete metric called *description length*. The description length of an $n \times m$ matrix A obtained by an $n \times k$ matrix B and an $k \times m$ matrix C matrix is defined as follows:

$$L(A,B,C) = L(n) + L(m) + L(k) + L(B) + L(C) + L(B \otimes C - A) \quad (5)$$

where $L(*)$ denotes the description length. A detailed discussion on the calculation of the description length can be found in [4]. Inspired by this work, we introduce the Multiple Factor Optimization (MFO) RMP.

Definition 10 (MFO RMP)**.** *Given a set of users U, a set of permissions $PRMS$ and a user-permission assignment UPA, find a set of roles R, a user-role assignment UA and a role-permission assignment PA minimizing*

$$L(M(UPA), M(UA), M(PA)) \quad (6)$$

where $L(M(UPA),M(UA),M(PA))$ denotes the description length of the representation of $M(UPA)$ using $M(UA)$ and $M(PA)$.

III. Motivation

Over the years several organizations are migrating to RBAC as a practical solution to manage authorizations in their IT systems. This has stimulated the design of several (tool-supported) solutions to automatically extract RBAC policies from access control policies expressed in term of users. These solutions, called role mining, require the set of permissions assigned to users to be retrievable and complete. However, retrieving the complete set of permissions within an organization is challenging. Next we present a scenario to illustrate these issues in the healthcare domain.

Example 1. *Consider a local healthcare center, such as a rehabilitation center, where patients of a small region are treated. The healthcare center offers various treatments, such as COPD rehabilitation. Patient and other administrative information is stored across the IT and medical infrastructures of the healthcare center. To access patient information the staff of the hospital have to use different applications depending on the task to be performed. For instance, doctors use a specific application to manage patients' medical information and another to retrieve lab results. Moreover, the healthcare center has administrative personnel for financial management and to make appointments with patients. Such personnel use an independent application to perform their tasks.*

The healthcare center desires to migrate to a modern Electronic Health Record (EHR) system that employs RBAC to regulate the access to patient information. However, the IT experts of the organization are challenged by the fact that they could not retrieve the complete set of permissions assigned to users. Most permissions are actually hard coded in different applications or stored across a number of legacy systems of the healthcare center. We stress that this situation is quite common in organizations with an outdated IT infrastructure.

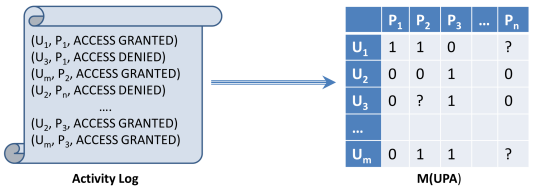


Fig. 1: *UPA* matrix representation of an activity log

The IT system of the healthcare center records the actions performed by users in activity logs for accountability purposes. As shown in Fig. 1, activity logs often contain the access requests for resources made by users along with the access decision (i.e., access granted or denied). Thereby, the activity logs collected from all the applications can be analyzed to determine the permissions assigned to users. Activity logs, however, may not provide a complete representation of the set of permissions assigned to users. In fact, it is unlikely that users will exercise all their permissions. This can be represented as a missing value *in the UPA (denoted by "?" in Fig. 1).*

Mining roles with incomplete knowledge is still an open issue. We argue that this issue should be addressed to provide organizations with practical solutions for migrating to an RBAC system.

IV. RMP Revisited

In this section we address the role mining problem with incomplete knowledge. First, we provide a formal representation of the activity logs. Next, we revisit existing role mining problems to consider missing values.

As discussed in Section III, the set of permissions assigned to users may be integrated in different applications making their retrieval and management difficult. In this work, we exploit the fact that the actions performed by users are often captured by the IT systems in activity logs.

Definition 11 (Activity Log)**.** *Let U be the set of users, OPS the set of operations, OBJ the set of objects and D the set access decisions. An activity log is a sequence of tuples $\langle u, op, obj, d \rangle$, where $u \in U$, $op \in OPS$, $obj \in OBJ$ and $d \in D$.*

Intuitively, an activity log records access requests along with the corresponding decision. The decision denotes whether a certain action was permitted or not.

The role mining problem with incomplete knowledge can be formalized by extending the RMPs defined in Section II. In this work, we focus only on the MinNoise RMP and MFO RMP with missing values. The MinNoise RMP is interesting in practice as it provides the best solution with respect to the approximation error, given an estimation of the number of roles. On the other hand, the MFO RMP provides the best

$\oslash$	**0**	**?**	**1**
0	0	1	1
?	0	1	0
1	1	1	0

(a) Dissimilarity operator

$\wedge$	**0**	**?**	**1**
0	0	0	0
?	0	?	?
1	0	?	1

(b) AND

$\vee$	**0**	**?**	**1**
0	0	?	1
?	?	?	1
1	1	1	1

(c) OR

Fig. 2: Ternary Matrix Operators

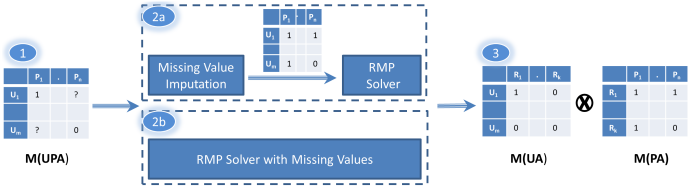


Fig. 3: Solving RMP with missing values

		MV Imputation		Native MV Support
		Zero	k-NN	
MinNoise RMP	ASSO	✔	✔	TMF
MFO RMP	PANDA	✔	✔	✘
	NASSAU	✔	✔	✘

TABLE I: Studied approaches for RMP with missing values

solution with respect to a wide range of aspects, without any additional knowledge required. We leave the extension of the other RMPs for future work.

An activity log can be represented using an $m \times n$ *UPA* matrix, where m and n are the number of users and operations within the system respectively (see Fig. 1). In this representation each entry of the activity log corresponds to a cell in the *UPA* matrix. A cell in the *UPA* matrix is set to 1 if the operation was permitted for the user or to 0 if it was denied. Since the activity log can be incomplete, some cells might not have a value; we explicitly mark them with "?". Now we formally define the MinNoise and MFO RMPs with missing values based on the *UPA* matrix representation of the activity log.

Definition 12 (MinNoise RMP with missing values)**.** *Given an activity log, let UPA be the user-permission assignment obtained from the log. Given a number of roles k, find a set of k roles R, a user-role assignment UA and a role-permission assignment PA, minimizing*

$$||M(UA) \otimes M(PA) \oslash M(UPA)||_1 \tag{7}$$

where $\oslash$ is the matrix dissimilarity operator as defined in [7] and shown in Fig. 2a.

The matrix dissimilarity operator $\oslash$ extends the matrix difference operator $-$ to handle missing values. The intuition behind this operator is to replace the missing values in the *UPA* with a known value in its factors (i.e., *UA* and *PA*).

Definition 13 (MFO RMP with missing values)**.** *Given an activity log, let UPA be the user-permission assignment obtained from the log. Find a set of roles R, a user-role assignment UA and a role-permission assignment PA minimizing*

$$L(M(UPA), M(UA), M(PA)) \tag{8}$$

where $L(M(UPA),M(UA),M(PA))$ denotes the description length of the representation of $M(UPA)$ using $M(UA)$ and $M(PA)$.

V. Role mining with missing values

Existing role mining solutions do not address the problem of missing values. In this section we discuss practical approaches to solve this problem for the MinNoise and MFO RMPs.

A. Towards a practical solution for RMP with missing values

To address the missing value problem within the RMP domain we investigate the use of different approaches to mine an RBAC policy based on incomplete information. In particular, given a *UPA* with missing values, the aim is to extract the *PA* and *UA* matrices that satisfy the optimization objective defined by the RMP (see Definitions 12 and 13).

Two main approaches can be adopted to address the missing value problem within the role mining context (Fig. 3). The first approach is to obtain a complete *UPA* and then apply an existing role mining solution to the obtained matrix; the second is to use a specialized RMP solution with intrinsic support for reasoning over incomplete knowledge.

In the first case, a missing value imputation step (referred to as *data cleansing*) can be used to replace missing values existing in the *UPA* matrix and, thus, obtain the complete set of permissions assigned to users. It is worth noting that the missing value imputation process is independent from the particular RMP problem (e.g., minNoise or MFO RMPs). The missing value imputation step replaces missing values in the *UPA* matrix with the most appropriate value according to a given metric, based on the relations among the data in the matrix. Different imputation approaches have been proposed. A naïve approach, which is often used in role mining, is to replace missing values with '0', i.e. it assumes that access was denied. We refer to this data cleansing approach as *Zero imputation*. The imputation of missing values is addressed in other fields such as data mining and recommender systems, although most of the existing proposals cannot be directly applied to Boolean data such as *UPA* matrices. In this work, we focus on k-NN missing value imputation [8], one of the most popular data cleansing method. The basic intuition behind k-NN imputation is to replace a missing value based on its relation with its nearest neighbors (e.g., neighboring rows, neighboring cells) according to a distance/similarity metric.

In this work, we investigate: *a)* the application of missing value imputation solution together with existing RMP solutions and *b)* the application of specialized solutions for the RMP problem which are able to extract roles from *UPA* matrices with missing values. First, we discuss the application of our approach for the minNoise RMP and later for the MFO RMP. A summary of our investigation is shown in Table I.

As shown in [3], minNoise RMP is equivalent to the BMF problem. In BMF the input data (represented as a Boolean matrix) is decomposed into two factor Boolean matrices.

Definition 14 (Boolean Matrix Factorization)**.** *Given an $m \times$*

n Boolean matrix C and a positive integer k, find an $m \times k$ Boolean matrix S, named usage matrix, and a $k \times n$ Boolean matrix B, named basis matrix, that minimize

$$||S \otimes B - C||_1 \tag{9}$$

The notion of factors in BMF is equivalent to the notion on roles in role mining. Thus, approaches to solve the BMF problem can be employed for the minNoise RMP. In this work we employ ASSO [6], one of the most popular and best performing algorithms for BMF. ASSO is inspired by association rule mining approaches [9] and uses the correlations between rows to define candidate factors, from which the final factors are selected. ASSO, however, cannot be directly applied to the minNoise RMP with missing values as it works only with complete data. Therefore, the data cleansing step has to be performed before its application.

To the best of our knowledge, only one BMF algorithm supports reasoning over missing values and can be directly applied to solve the minNoise RMP with missing values. In particular, the work in [7] introduces Ternary Matrix Factorization (TMF) and shows that BMF is a specialization of it. TMF allows the existence of missing value in the basis matrix B and uses a specialized dissimilarity metric to assist the missing value imputation. Therefore, we argue that the minNoise RMP with missing values and TMF with the specialized dissimilarity metric are equivalent and that it can be directly applied to discover roles from an incomplete *UPA* matrix.

Definition 15 (Ternary Matrix Factorization)**.** *Given a ternary $n \times m$ matrix C with elements $c_{ij} \in \{0,1,?\}$ ($i \in \{1,\ldots,n\}$, $j \in \{1,\ldots,m\}$) and a positive integer k, find a Boolean $n \times k$ usage matrix S and a ternary $k \times m$ basis matrix B that minimizes*

$$||(S \triangle B) \oslash C||_1 \tag{10}$$

where $\triangle$ denotes the ternary matrix product and $\oslash$ the dissimilarity metric (see Fig. 2). The ternary matrix product $\triangle$ is similar to the Boolean matrix product $\otimes$ where the conjunction $\wedge$ and disjunction $\vee$ operators are extended to deal with missing values as shown in Fig. 2.

As discussed in Section II, the MFO RMP is related to the Minimum Description Length Boolean matrix factorization problem (MDL BMF) [4]. To the best of our knowledge, there do not exist approaches that natively support missing values for this problem. Therefore, we focus on the application of existing MDL BMF solutions preceded by a data cleansing step. In particular, we adopt PANDA [10] and NASSAU [11], two of the best performing algorithms for the MDL BMF problem. PANDA is a unifying framework to efficiently mine the top-k binary patterns, a problem similar to BMF. In particular, it offers a range of different metrics that can be used in the optimization process, including description length. The idea underlying PANDA is to iteratively find dense cores (i.e., a submatrix full of 1s) in the dataset and then extend the core with additional rows and columns. NASSAU, on the other hand, is an algorithm proposed to explicitly solve the MDL BMF. The key novelty of NASSAU is that, differently from most of the existing algorithms including PANDA, it periodically revises the previous choices (i.e., selected cores in the factorization) in order to decrease the description length.

RMP Problem	WSC Weights
RMP	$(1,0,0,\infty)$
δ-Approx RMP	$(1,0,0,1)$
Minimal Edge RMP	$(0,1,1,\infty)$
Min-Noise RMP	$(0,0,0,1)$
MFO RMP	$(1,1,1,1)$

TABLE II: Relation between RMP problems and WSC weights

It is worth noting that the MDL problem is based on the minimization of the description length, a metric based on information theory. Although this metric is well suited for data mining, it does not fit the role mining objectives. Here, we are interested in the number of roles and the approximation error rather than in the description length (i.e., the number of bits of the representation) of these entities. To this end, we argue that a more suitable metric for the evaluation of RMP solutions, than the description length, should be used. We discuss this in the following section.

B. RMP solution evaluation

Given a user-permission assignment *UPA*, many RBAC states can be consistent with it. Each RMP employs a particular criterion to evaluate the quality of solutions. For instance, the number of roles is used to evaluate the quality of the solutions for the basic RMP and δ-Approx RMP. On the other hand, the approximation error δ is used to evaluate the quality of solutions for the MinNoise RMP. A commonly used generic metric to measure the quality of an RBAC state is the Weighted Structural Complexity (WSC) [12], [13]. This metric measures the size of an RBAC state as a linear combination of various measures, such as the number of role assignments. Based on the WSC metric it is possible to select the best RBAC state that is consistent with an access control configuration.

Definition 16 (Weighted Structural Complexity)**.** *Given a vector of weights $W = (w_r, w_u, w_p, w_d)$ s.t. $\forall w_i \in \mathbb{R}$, the weighted structural complexity $wsc(RS, W)$ of an RBAC state $RS = (R, UA, PA)$ derived from an initial UPA is*

$$\begin{aligned} wsc(RS,W) = w_r|R| + w_u|M(UA)| + w_p|M(PA)| \\ + w_d|M(UA) \otimes M(PA) - M(UPA)| \end{aligned} \tag{11}$$

where $|\cdot|$ represents the size of the corresponding set (L_1 norm is used for matrices).

As shown in Table II, the WSC can be used to denote the objectives and minimization goals of most RMPs [13]. For instance, the basic RMP aims to minimize the number of roles, which can be obtained by setting $W = (1,0,0,\infty)$. Recall that, in this problem, the derived RBAC state should be 0-consistent with the initial *UPA*; this is obtained by setting $w_d = \infty$. Similarly, the minimization objective of the MinNoise and MFO RMP can be met by setting $W = (0,0,0,1)$ and $W = (1,1,1,1)$ respectively. Note that arithmetic operations

involving ∞ are handled as follows: $0+\infty=0$ and $x+\infty=\infty$ where $x \in \mathbb{R} \cup \{\infty\}$.

Although the WSC metric considers the most important aspects of a solution to the role mining problem, the different aspects do not have an equivalent impact on the value of the metric. This can result in a biased evaluation of an RBAC state.

Example 2. *Given an initial* 10×30 *UPA matrix and an RBAC state* $RS = (R, UA, PA)$ *with* 4 *roles, one can observe that the* 10×4 *UA matrix has a lower impact than the* 4×30 *PA matrix when using the* L_1 *norm. This is because the* L_1 *norm of UA ranges within* $[0, 40]$ *while the* L_1 *norm of PA ranges within* $[0, 120]$. *Moreover, both these aspects have a larger impact than the number of roles.*

To address the issues related to the evaluation of RBAC states using the WSC metric we propose to normalize the different aspects. In particular, the aspects expressed as matrices (i.e., *UA*, *PA*, *UPA*) can be normalized using the matrix size (i.e., the product of its dimensions). For the role aspect, one may use the maximal number of possible roles as normalization factor, where the set of potential roles is equal to the powerset of the set of permissions. However, using the cardinality of the powerset as the normalization factor is not appropriate due to the potentially large number of the possible combinations of permissions, which can reduce the impact of the role aspect. To this end, we use the number of users as the normalization criterion for the role aspect. The intuition underlying this choice is that an RMP will never result in an RBAC state in which the number of roles is greater than the number of users. Next, we define the Normalized Weighted Structural Complexity.

Definition 17 (Normalized Weighted Structural Complexity)**.** *Given a vector of weights* $W = (w_r, w_u, w_p, w_d)$ *s.t.* $\forall w_i \in \mathbb{R}$, *the normalized weighted structural complexity* $nwsc(RS, W)$ *of an RBAC state* $RS = (R, UA, PA)$ *derived from an initial* $m \times n$ *UPA is*

$$nwsc(RS,W) = w_r \frac{|R|}{m} + w_u \frac{|M(UA)|}{m \times |R|} + w_p \frac{|M(PA)|}{|R| \times n} + w_d \frac{|M(UA) \otimes M(PA) - M(UPA)|}{m \times n} \quad (12)$$

where $|\cdot|$ *represents the size of the corresponding set (*L_1 *norm is used for matrices).*

As discussed in Section V-A, the description length might not be a suitable metric to correctly express the objectives of the MFO RMP problem. We propose to replace the description length with the WSC and nWSC metrics, using $W = (1, 1, 1, 1)$. To this end, we have modified PANDA and NASSAU to accommodate the WSC and nWSC metrics as the minimization criteria. We study the impact of these modifications on the performance of MFO RMP algorithms in the next section.

VI. Experiments

In this section we evaluate the performance of the approaches presented in Section V. In particular, we evaluate the quality of the solutions for the MinNoise and MFO RMPs with missing values using both synthetic and real-life datasets.

For the MinNoise RMP we evaluate the performance of ASSO in combination with two data cleansing methods, namely Zero and k-NN imputation. In addition, we compare the results to the solutions generated by TMF. To measure the quality of results we assess the approximation error of the mined *UPA* matrix for a varying number of roles.

For the MFO RMP we evaluate the performance of PANDA and NASSAU combined with both Zero and k-NN imputation. Moreover, we study the impact of the WSC and nWSC metrics on the quality of solutions obtained for the MFO RMP. We measure the quality of the results by assessing the approximation error of the mined *UPA* matrix and the number of mined roles.

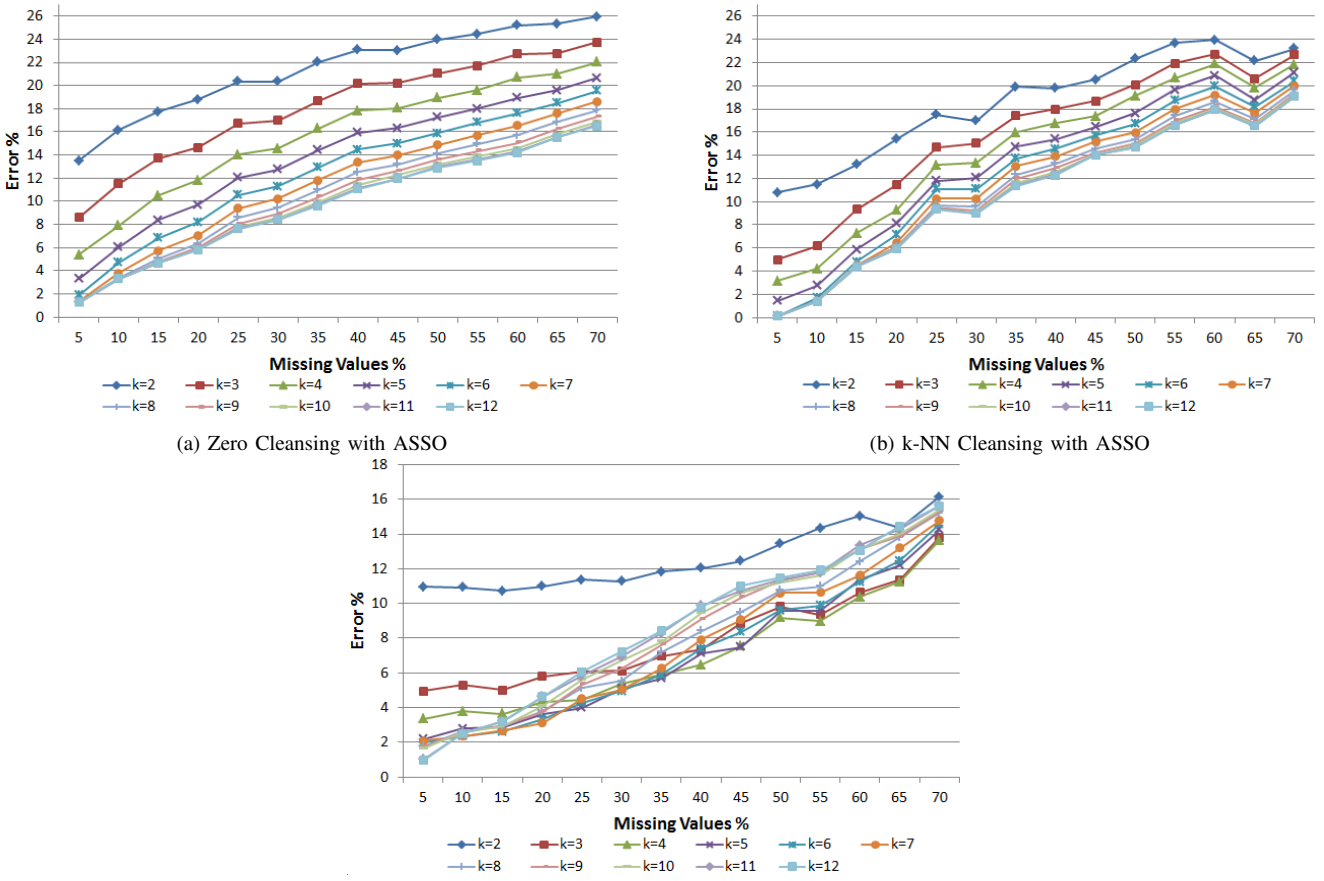
A. Synthetic Data

In this section we present the experiments conducted with synthetic data.

Experiment Setting: For the evaluation of the approaches for the RMP with missing values we constructed a ground truth RBAC policy based on Example 1. This policy consists of 6 roles, representing different positions within the healthcare center, such as doctors, specialists, nurses and administrative personnel. The policy contains 15 different permissions representing different types of access (e.g., read, modify) on different types of data (e.g., medical, demographical). Roles are assigned to 15 users of the system. Based on this RBAC policy we derived the corresponding *UPA* matrix, which is used as the ground truth for the evaluation. We defined 14 different sets of incomplete *UPA*s varying the percentage of missing values from 5% to 70% (with 5% interval). Each set consists of 25 incomplete *UPA*s generated from the ground truth *UPA*, for a total of 350 *UPA*s containing missing values.

The synthetic dataset was used to conduct two sets of experiments. In the first set we evaluate the performance of approaches for the MinNoise RMP with missing values. To this end, we varied the number of roles ranging from 2 to 12 roles. In the second set of experiments we evaluate the performance of the approaches for the MFO RMP with missing values.

Results: The results of the first set of experiments are shown in Fig. 4. These graphs present the average approximation error per each set of incomplete *UPA* matrices with respect to the ground truth *UPA* matrix for different numbers of roles (k). As expected, the results show an increasing approximation error as the ratio of missing values increases. By comparing Fig. 4a and Fig. 4b, we can observe that the use of k-NN for data cleansing results in 2% lower error ratio compared to Zero imputation on average. Moreover, the number of roles has a significant impact on the quality of the solution. In particular, the approximation error decreases as the number of roles increases. However, as shown in Fig. 4b, the approximation error tends to converge after a certain value of k, meaning that a better solution cannot be found regardless of the number of roles. This trend is even more obvious as the percentage of missing values increases.



(a) Zero Cleansing with ASSO

(b) k-NN Cleansing with ASSO

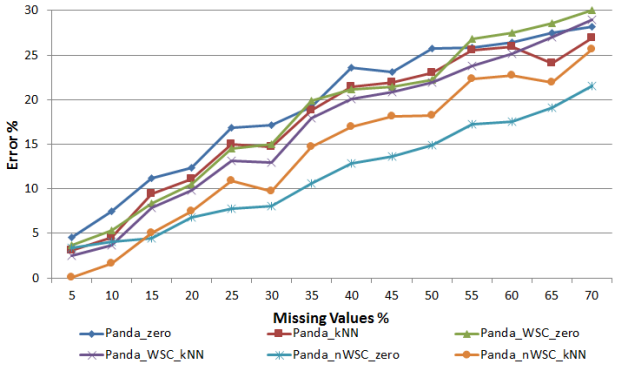
(c) TMF

Fig. 4: MinNoise RMP: approximation error

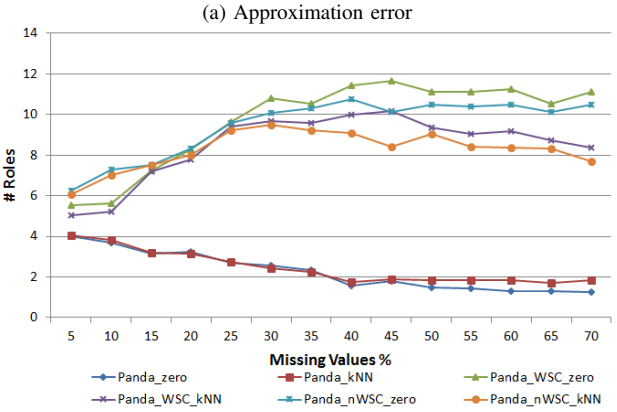
Fig. 4c shows that TMF presents the lowest error when the value of k is close to the number of roles in the ground truth RBAC policy. Interestingly, for a missing value ratio lower than 40% the lowest error is obtained when k is over but close to 6 roles, while for a missing value ratio higher than 40% the lowest error is obtained when k is slightly lower than 6 roles. In general, TMF performs better than the ASSO-based solutions mainly due to its native support for incomplete knowledge. A notable exception is when the missing value ratio is below 10%. In this case, ASSO combined with k-NN imputation presents almost 0% error.

The results of the second set of experiments are shown in Fig. 5 and Fig. 6. The first figure shows the average approximation error with respect to the ground truth policy and the average number of roles mined by PANDA, while Fig. 6 shows the average approximation error and average number of roles mined by NASSAU. The results show that the unmodified versions of PANDA and NASSAU provide the lowest quality results in most of the cases and fail to discover the roles as defined in the original RBAC policy, with PANDA providing slightly better performance than NASSAU. In particular, NASSAU mined from 1 to 3 roles achieving the best results when the missing value rate is low (Fig. 6b). Moreover, the results show that the imputation of missing values does not have a significant impact on the quality of the solutions. In addition, the unmodified versions of PANDA and NASSAU provide the highest approximation error in majority of the cases. These poor results can be justified by the fact that the description length in not a suitable metric for RMP. This intuition is confirmed by the results obtained using the WSC and nWSC variants of PANDA and NASSAU. In particular, the WSC variants of the algorithms have a lower approximation error compared to their MDL counterparts. Moreover, there is a major improvement in the number of mined roles for the WSC variant of PANDA (Fig. 5b).

Although the application of the WSC metric to the MFO RMP is beneficial in most of the cases, the obtained gains is marginal especially for NASSAU. On the other hand, the nWSC variants of both PANDA and NASSAU show a significant improvement. In particular, the nWSC variants provide the lowest approximation error among their counterparts. For instance, the approximation error for PANDA varies from almost 0% for 5% of missing values to 22% for 70% of missing values. Moreover, the nWSC metric provides a better estimation of the number of roles. This can be clearly observed in the case of NASSAU for which the use of the nWSC metric lead to discover from 5 to 6 roles (Fig. 6b), which is close to the number of roles in the ground truth RBAC policy.

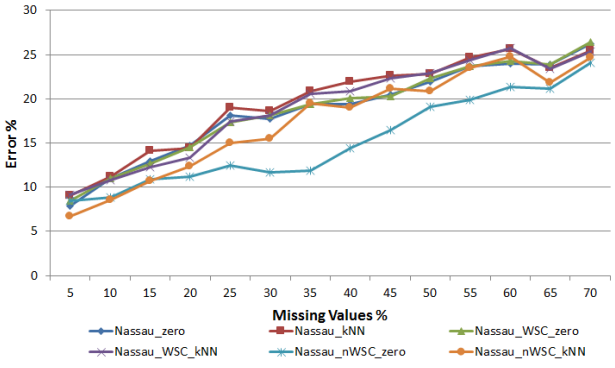


(a) Approximation error

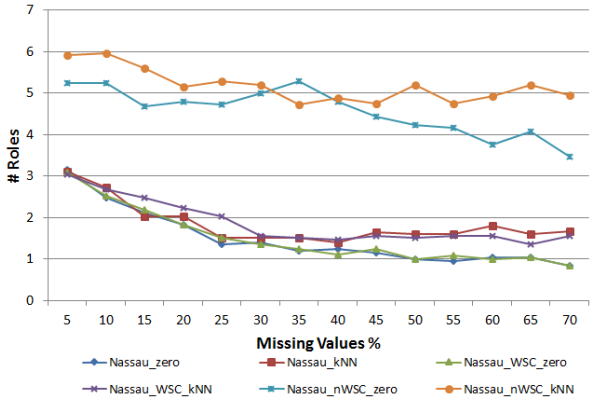


(b) Number of roles

Fig. 5: MFO RMP using PANDA



(a) Approximation error



(b) Number of roles

Fig. 6: MFO RMP using NASSAU

B. Real Data

We also evaluated the approaches discussed in this work using a real dataset.

Experiment Setting: To evaluate the approaches for RMP with missing values we used a real dataset obtained from SAFAX [14], a novel XACML-based framework that offers authorization as a service. SAFAX employs an access control policy consisting of 57 permissions allocated to 6 roles to regulate the use of the authorization service itself. Based on this policy, we derived the ground truth *UPA* matrix. The *UPA* used for the experiments was obtained from the activity log of SAFAX, which consists of 4309 access requests made by 67 users along with the corresponding access decision. The constructed *UPA* matrix contains 80.75% missing values.

In the experiments we first evaluate the performance of the approaches for the MinNoise RMP with missing values; then we evaluate the performance of the approaches for the MFO RMP with missing values. We also studied the impact of the WSC and nWSC metrics on the solutions for the MFO RMP.

Results: The results for the minNoise RMP are shown in Fig. 7. The figure shows the approximation error of each approach with respect to the ground truth *UPA* matrix for a varying number of roles (k). Similarly to what observed in the results for the synthetic dataset, the approximation error for ASSO decreases as the number of roles increases until it converges. However, in this case k-NN imputation performs

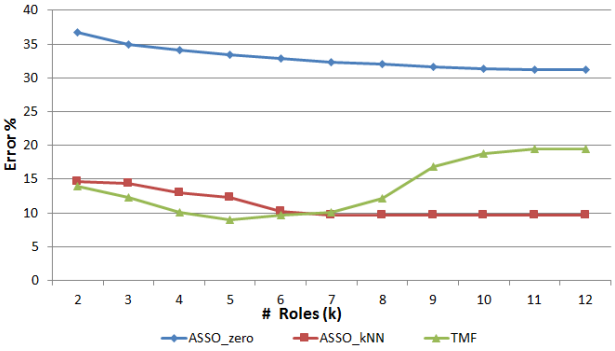


Fig. 7: minNoise RMP: approximation error

significantly better than Zero imputation. These results can be justified by the fact that the ground truth *UPA* obtained from the SAFAX policy has a higher ratio of 1's, thus resulting in a higher approximation error when Zero imputation is used.

Fig. 7 shows that TMF provides the lowest approximation error when the number of roles is close to the number of roles in the SAFAX policy. In particular, the lowest approximation error is obtained when the number of roles was set to 5. However, for k larger than 8 ASSO with k-NN presents a significantly lower approximation error than TMF. Therefore, we conclude that TMF should be used when a good estimation of the number of roles is known; otherwise, ASSO is preferable.

		Description Length		WSC		nWSC	
		k-NN	Zero	k-NN	Zero	k-NN	Zero
PANDA	**# Roles**	2	3	3	4	4	8
	Error %	25.97	36.55	19.2	34.2	18.17	32.7
NASSAU	**# Roles**	13	3	3	5	4	6
	Error %	10.7	39.8	10.2	32.6	9.1	30

TABLE III: MFO RMP using PANDA and NASSAU

The results for MFO RMP are shown in Table III. The table shows the number of roles and approximation error with respect to the ground truth *UPA* matrix for PANDA and NASSAU approaches. In general, the results with the real dataset are similar to the ones with the synthetic dataset. In particular, we can observe that the unmodified versions of PANDA and NASSAU provide lower quality results in terms of both number of roles and approximation error than their modified counterparts. Especially for PANDA, we can observe a drop of more than 5% in the approximation error when using WSC or nWSC while providing a better estimation of the number of roles. Differently from the experiments presented in Section VI-A, the approach used for missing value imputation has a significant impact on the quality of the extracted RBAC policy. In particular, Zero imputation leads to a significantly higher approximation error (more than 30% in every case). Based on these results, we can conclude that the use of nWSC is beneficial for both PANDA and NASSAU and provides the lowest approximation error and the best estimation of the number of roles compared to the WSC and unmodified algorithms.

VII. Related Work

The definition of roles is crucial for the deployment and adoption of an RBAC system [2]. Existing solutions can be classified into two main categories: role engineering and role mining. Role engineering aims to discover roles by analyzing organization policies and business processes. Early work on role engineering elicits roles from use-cases [15] while newer approaches offer more sophisticated methods such as deriving roles from business processes [16], [17] or by employing UML [18], [19]. The major disadvantage of role engineering is that the human factor has significant importance and thus the elicitation of roles cannot be automated.

This issue is typically addressed by role mining. Role mining aims to discover roles from the set of permissions assigned to users and typically relies on data mining techniques, such as subset enumeration [20], clustering [21], [22], graph theory [23], [24] and BMF [3], [25]. Vaidya et al. [3] propose a formalization of the role mining problem using matrix notation and present several variants of the RMP to meet different mining objectives such as the minimization of the number of roles or approximation error. The approach to these problems is, however, NP-complete and thus only approximated solutions can be found [3]. Moreover, Vaidya et al. investigate the relation of the role mining problem with other known problems in the literature. For instance, they show that the basic RMP problem can be related to the Minimum Tiling problem and the MinNoise RMP to the Discrete Basis Problem (i.e., BMF).

Role mining approaches usually require the complete set of permissions assigned to users as input in order to mine roles. However, it is difficult to meet this requirements in practice; information about the permissions that users have is often fragmented and not directly accessible. Existing solutions address this issue by treating missing values as a 'deny', i.e. they adopt Zero imputation to impute missing values. However, as shown in Section VI-B, Zero imputation may provide unsatisfactory results in real systems. To the best of our knowledge, RMP with missing values has not been addressed in the literature. A few proposals [20], [26] investigate the robustness of role mining solutions to noise. The notion of noise, however, is different from the notion of missing values as noise refers to permissions incorrectly assigned to users.

Reasoning over missing values is addressed in other fields, such as data mining and recommender systems, where data are often incomplete. To tackle this problem, data cleansing solutions have been proposed to replace missing values in a matrix with the most appropriate value according to a given metric. However, most of the data cleansing solutions focus on real or non-negative numbers and cannot be directly applied to Boolean matrices. For Boolean matrices the most commonly used data cleansing algorithms are k-nearest neighbors (k-NN) [8] and local least squares imputation [27]. Another approach is KRIMP-minimization [28], which relies on information theory to impute the missing values in a matrix. KRIMP-minimization employs KRIMP to provide descriptions of Boolean matrices through compression of the data using frequent itemsets. The intuition behind this approach is that the best values for replacing a missing value are the ones that compress the data the most.

The major drawback of data cleansing methods is that they replace missing values only by examining the relation of the existing data without taking into account any problem-specific objective, such as the minimization of the number of roles or the approximation error for the RMP case. A notable exception among data cleansing methods is ABBA [29], an adaptive bicluster-based approach that leverages identifiable patterns (biclusters) within the data to replace missing values in a Boolean matrix. In particular, this approach exploits the similarity between the notion of biclusters and the one of candidate roles in the RMP [29]. ABBA uses an iterative approach to incomplete *UPA* matrices in order to minimize the number of maximal biclusters and, thus, the number of candidate roles. Such an approach, however, can lead to a large approximation error and to assign more permissions than necessary. Apart from data cleansing approaches, the missing value problem has been also addressed in the Boolean matrix factorization field [7], which is related to the MinNoise RMP problem. In particular, these approaches aim to impute missing information considering the BMF problem objective, which is the minimization of the approximation error. To the best of our knowledge, this work is the first that have studied the applicability of these approaches to role mining.

VIII. Conclusion

Many organizations are migrating to RBAC as a practical solution to protect sensitive information. Role mining has been proposed to discover roles from the (complete) set of permissions assigned to users. However, the complete set of permissions assigned to users is often unavailable, especially for organizations with an outdated IT infrastructure. In this work, we have investigated the problem of role mining with incomplete knowledge. In particular, we have provided a formal representation of the MinNoise and MFO RMPs with missing values and discussed practical approaches for these problems. Moreover, we have studied metrics to evaluate the mined policies and applied them to the MFO RMP. The studied approaches were evaluated using both synthetic and real datasets.

The quality of RBAC policies mined from logs depends on the quality of the log used to mine the roles. In particular, role mining assumes that the set of user permissions used to mine the roles reflects the permissions currently assigned to users. Although this assumption holds for the logs used in our experiments, it may not hold in general. For instance, a log can include permissions a user has accumulated over the years. To this end, a pre-processing phase may be required to identify which permissions are actually assigned to users according to their current job position.

This work has shown that intrinsic support for missing values can have a significant impact on the quality of the mined RBAC policy. For instance, our experiments showed that TMF provides very promising results for the MinNoise RMP with missing values. Thus, the development of solutions tailored to the RMP with missing values can provide increasing benefits to organizations aiming to migrate to RBAC by allowing them to extract more effective and accurate RBAC policies. In this work we also used existing data cleansing approaches to missing values. These approaches, however, are not specific to RMP. An interesting direction for future work is the design of data cleansing solutions tailored to the needs of RMP. Finally, future work includes the investigation of the missing value problem for the other RMP instances, such as the δ-Approx RMP.

Acknowledgments: This work has been partially funded by the ITEA2 projects FedSS (No. 11009) and M2MGrid (No. 13011), the EDA project IN4STARS2.0, and the Dutch national program COMMIT under the THeCS project.

References

[1] Research Triangle Institute, "Economic analysis of role-based access control," 2010.

[2] E. J. Coyne, "Role engineering," in *Proceedings of Workshop on Role-Based Access Control*. ACM, 1996, p. 4.

[3] J. Vaidya, V. Atluri, and Q. Guo, "The role mining problem: A formal perspective," *TISSEC*, vol. 13, no. 3, p. 27, 2010.

[4] P. Miettinen and J. Vreeken, "MDL4BMF: Minimum description length for Boolean matrix factorization," *TKDD*, vol. 8, no. 4, p. 18, 2014.

[5] J. Vaidya, V. Atluri, Q. Guo, and H. Lu, "Edge-RMP: Minimizing administrative assignments for role-based access control," *Journal of Computer Security*, vol. 17, no. 2, pp. 211–235, 2009.

[6] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila, "The discrete basis problem," *TKDE*, vol. 20, no. 10, pp. 1348–1362, 2008.

[7] S. Maurus and C. Plant, "Ternary matrix factorization," in *Proceedings of International Conference on Data Mining*. IEEE, 2014, pp. 400–409.

[8] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[9] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 207–216, 1993.

[10] C. Lucchese, S. Orlando, and R. Perego, "A Unifying Framework for Mining Approximate Top-k Binary Patterns," *TKDE*, vol. 26, no. 12, pp. 2900–2913, 2014.

[11] S. Karaev, P. Miettinen, and J. Vreeken, "Getting to know the unknown unknowns: Destructive-noise resistant boolean matrix factorization," in *Proceedings of SIAM International Conference on Data Mining*, 2014.

[12] I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. Calo, and J. Lobo, "Mining roles with semantic meanings," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2008, pp. 21–30.

[13] I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang, and J. Lobo, "Evaluating role mining algorithms," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2009, pp. 95–104.

[14] S. P. Kaluvuri, A. I. Egner, J. den Hartog, and N. Zannone, "SAFAX–An Extensible Authorization Service for Cloud Environments," *Frontiers in ICT*, vol. 2, p. 9, 2015.

[15] E. B. Fernandez and J. Hawkins, "Determining role rights from use cases," in *Proceedings of Workshop on Role-Based Access Control*. ACM, 1997, pp. 121–125.

[16] G. Neumann and M. Strembeck, "A scenario-driven role engineering process for functional RBAC roles," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2002, pp. 33–42.

[17] H. Roeckle, G. Schimpf, and R. Weidinger, "Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization," in *Proceedings of Workshop on Role-Based Access Control*. ACM, 2000, pp. 103–110.

[18] P. Epstein and R. Sandhu, "Towards a uml based approach to role engineering," in *Proceedings of Workshop on Role-Based Access Control*. ACM, 1999, pp. 135–143.

[19] D. Shin, G.-J. Ahn, S. Cho, and S. Jin, "On modeling system-centric information for role engineering," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2003, pp. 169–178.

[20] J. Vaidya, V. Atluri, J. Warner, and Q. Guo, "Role engineering via prioritized subset enumeration," *TDSC*, vol. 7, no. 3, pp. 300–314, 2010.

[21] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role mining–revealing business roles for security administration using data mining technology," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2003, pp. 179–186.

[22] J. Schlegelmilch and U. Steffens, "Role mining with ORCA," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2005, pp. 168–176.

[23] D. Zhang, K. Ramamohanarao, and T. Ebringer, "Role engineering using graph optimisation," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2007, pp. 139–144.

[24] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan, "Fast exact and heuristic methods for role minimization problems," in *Proceedings of Symposium on Access Control Models and Technologies*. ACM, 2008, pp. 1–10.

[25] H. Lu, J. Vaidya, V. Atluri, and Y. Hong, "Constraint-aware role mining via extended Boolean matrix decomposition," *TDSC*, vol. 9, no. 5, pp. 655–669, 2012.

[26] J. Vaidya, V. Atluri, Q. Guo, and H. Lu, "Role mining in the presence of noise," in *Data and Applications Security and Privacy XXIV*. Springer, 2010, pp. 97–112.

[27] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[28] J. Vreeken and A. Siebes, "Filling in the Blanks – KRIMP Minimisation for Missing Data," in *Proceedings of International Conference on Data Mining*. IEEE, 2008, pp. 1067–1072.

[29] A. Colantonio, R. Di Pietro, A. Ocello, and N. V. Verde, "ABBA: Adaptive bicluster-based approach to impute missing values in binary matrices," in *Proceedings of Symposium on Applied Computing*. ACM, 2010, pp. 1026–1033.