An Accurate and Scalable Role Mining Algorithm based on Graph Embedding and Unsupervised Feature Learning

Masoumeh Abolfathi, Zohreh Raghebi, Jafar Haadi Jafarian, Farnoush Banaei-Kashani
University of Colorado, Denver
{masoumeh.abolfathi, zohreh.raghebi, haadi.jafarian, farnoush.banaei-kashani}@ucdenver.edu

Abstract

Role-based access control (RBAC) is one of the most widely authorization models used by organizations. In RBAC, accesses are controlled based on the roles of users within the organization. The flexibility and usability of RBAC have encouraged organizations to migrate from traditional discretionary access control (DAC) models to RBAC. The most challenging step in this migration is role mining, which is the process of extracting meaningful roles from existing access control lists. Although various approaches have been proposed to address this NP-complete role mining problem in the literature, they either suffer from low scalability, or present heuristics that suffer from low accuracy. In this paper, we propose an accurate and scalable approach to the role mining problem. To this aim, we represent user-permission assignments as a bipartite graph where nodes are users and permissions, and edges are user-permission assignments. Next, we introduce an efficient deep learning algorithm based on random walk sampling to learn low-dimensional representations of the graph, such that permissions that are assigned to similar users are closer in this new space. Then, we use k-means and GMM clustering techniques to cluster permission nodes into roles. We show the effectiveness of our proposed approach by testing it on different datasets. Experimental results show that our approach performs accurate role mining, even for large datasets.

1. Introduction

Organizations use access control models to decrease the risk of unauthorized access to systems, data, and resources.Meanwhile, scalable access control management, especially for large or even medium-sized organizations, is a challenging problem due to the size and variety of users and permissions.

Role-Based access control (RBAC) is an enterprise-oriented access control model which is widely used by organizations, enterprises, and governmental entities. According to a study by IBM, RBAC is ”creating both a valid Return On Investment (ROI) and driving better control over the assets of an organization” [1]. RBAC is broadly used in access control management of e-government systems [2], and many policy-based e-government security systems use this model to ensure maximum protection of critical data in their systems.

In RBAC, users’ access policies are assigned to them based on roles, where each role is essentially a set of permissions. Deploying RBAC in an organization critically depends on defining roles, which are a functional links between users and permissions. Thus, the first step towards using RBAC in an organization is to identify an appropriate set of roles. Since many organizations already have user-permission assignments defined as discretionary access control (DAC) lists, it makes sense to identify roles from this existing information. This process, known as role mining, is one of the critical steps for successful RBAC adoption in enterprises. The role mining problem is shown to be NP-complete [3]. Effective and efficient role mining, especially for large or even medium-sized organizations, is very challenging due to the high number of users, permissions, and possible roles.

Many approaches have been proposed in the literature for efficient role mining, using different methodologies and metrics [4, 5, 6, 7, 8, 9, 10]. However, existing body of work on role mining suffers from two main shortcomings: first, existing approaches define optimal role mining based on unrealistic metrics such as minimizing the number of extracted roles. However, these solutions do not strive to capture the contextual and structural correlation among users and permissions in order to define roles that potentially map to the existing organizational roles of users. Second, the role mining problem is NP-complete and meanwhile the number of users and permissions in organizations are very large, usually in the order of several thousands. As a result, existing approaches either rely on heuristics that

URI: https://hdl.handle.net/10125/63981
978-0-9981331-3-3

have accuracy issues [7, 10] or are not scalable [9].

In this paper, we propose a scalable and context-aware role mining technique based on graph embedding and unsupervised feature learning. Our proposed method performs accurate role mining for organizations with thousands of users and permissions. To capture and learn the structural similarity of permissions, we use an efficient random walk sampling based approach in order to extract roles which are meaningful in the context of the organization.

For efficient and context-aware role mining, first we convert access control list to a bipartite graph called "user permission graph" and then, we use clustering to find subsets of similar nodes and group them together as potential roles. However, to address scalability, instead of performing clustering on the original graph, we use an efficient graph embedding method to represent data in a new low-dimensional space. We leverage the properties of user-permission graph to improve the performance of our graph embedding solution. This dimension reduction significantly reduces the computational cost of role mining. Then, a clustering algorithm is applied in a low-dimensional space to extract candidate roles. Finally, a prune and refine algorithm is developed to verify the final roles and cover the corner cases. Our proposed method consists of the following steps:

- Constructing a graph of user permission list: We define the user-permission graph based on the existing access control list (DAC-list).

- Embedding the graph into a new low-dimensional space: Using random walk to capture similarity between nodes in a bipartite graph, we embed the graph into a new low-dimensional space.

- Clustering: Using k-means and GMM clustering methods, the permission nodes in the embedded graph are clustered as a role. In this step, we tune the learning parameters by using Elbow and Silhouette analysis scores to achieve the best performance of the built model.

- Prune and refine: To cover all corner cases and to mitigate the possible security impacts, a heuristic algorithm is proposed. In this step, we extract the final set of roles by pruning the candidate roles from previous step.

To evaluate the proposed approach, we run several experiments on a number of datasets of different sizes. For measuring the accuracy of the model, Jaccard similarity metric is used. We show that on the smaller datasets, our algorithm achieves 100% accuracy even without pruning. On larger dataset, pruning is needed. Even without pruning we achieve 67% accuracy. We have 100% accuracy with pruning.

The rest of this paper is organized as follows. In Section 2, we review the preliminaries and background information needed to establish an understanding of the problem domain. Section 3 discusses the related work. We present our proposed role mining approach in Section 4. We report the experimental results in Section 5. In Section 6, we discuss the GMM clustering and other similarity measures in graph embedding. Finally, in Section 7, we conclude the paper with future research directions.

2. Background and Preliminaries

In this section, we review the notions of basic RBAC, and terminologies in graph embedding.

2.1. Overview of RBAC Definition and Terminology

In RBAC, "role" is defined as a set of permissions on resources. New permissions can be granted to or revoked from roles as needed. A permission is an access right to an object in the system. The RBAC model is based on two key relations. User to role assignment (UR) and role to permission assignment (RP) relations. Both are many-to-many relations. A user can have many roles, and a role can be assigned to many users. Likewise, a role can include many permissions, and the same permission can be granted to many roles. Finally, it is a user who uses permissions. The placement of a role as a mediator to enable a user to exercise a permission provides more simplification on access control management than does directly assigning users to permissions [11].

The following definitions formalize the above discussion.

- U, R, P (users, roles, and permissions)

- $UR \subset U \times R$: a many-to-many user to role assignment relation

- $RP \subset R \times P$: a many-to-many role to permission assignment relation

- $UP \subset U \times P$: a many-to-many users to permission assignment relation

- $R \in \mathcal{P}(P) - \emptyset$ (where $\mathcal{P}$ is power set)

2.2. Graph Embedding

The goal of graph embedding is to represent a graph in a new low-dimensional space which is structurally in equivalence with the original input graph. There exist several traditional approaches to learning low-dimensional graph representations, including multidimensional IsoMap [12], LLE [13], and Laplacian Eigenmaps [14]. These works apply dimension reduction techniques such as singular value decomposition (SVD) or principal component analysis (PCA) on the graph adjacency matrix or graph Laplacian matrix to obtain node representation in low dimension. However, the poor scalability of these approaches makes it difficult to be applied to large-scale graphs.

Recently, another line of research provides solutions to the representation learning of nodes in large-scale graphs with inspiration from neural language models (e.g., Skip-gram). DeepWalk [15] is one of the first works that presented an approach which transforms graph structure into several linear node sequences by using uniform random walks and then generates node representations by using Skip-gram model. LINE [16] addresses the graph embedding problem for all types of directed, undirected, weighted, and un-weighted graphs. LINE preserves the local and global graph structure by considering first-order proximity which refers to the local vicinity between every two vertices in the graph and the second-order proximity maintaining the general structure of the graph.

Node2vec [17] extended DeepWalk by employing biased random walks to learn node embeddings. The authors study the second-order random walk built based on the transition probability between nodes. Moreover, there are some follow-up works including GraRep [18] and Hope [19, 20] that use other similarity measures between vertices such as pagerank and katz to capture high-order proximity.

Here we formalize the above discussions. Let $G = (V, E)$ be a given bipartite graph, where V and E are the set of vertices (nodes) and edges, respectively. Particularly, bipartite graphs have the relationship between two types of entities. We define a representation as a mapping $\Phi : v \in V \to R^{|v|*d}$ that maps each vertex to a d-dimensional vector in the real space. Here, d is defined to specify the number of dimensions of our feature representation.

3. Related Work

To address the role mining problem, Vaidya proposes a method called *CompleteMiner* [4]. The main idea of this method is to use the concept of subset enumeration. In the first phase of the algorithm, by preprocessing the original data, roles are initialized from all possible existing permission combinations since each role is defined as a set of permissions. Then the set *InitRoles* is generated by considering the users who have the same permissions as a unique role set. In phase 2, CM enumerates all intersection sets between sets in *InitRoles* and adds them to *InitRoles*. Whereas the computational complexity of this algorithm is exponential, the authors made a notable improvement in their proposed algorithm by intersecting only pairs of *InitRoles*. The new approach is called *FastMiner* [4]. *FastMiner* is a heuristic approach which finds only the roles that are maximally common to any two users. The computational complexity of the algorithm reduces to $O(n^2m)$. However, this approach ignores some significant roles.

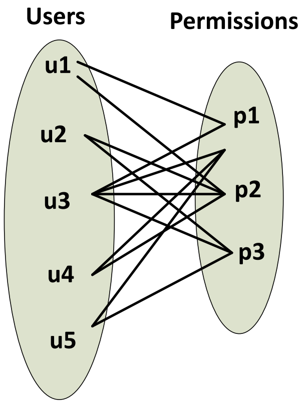

Figure 1. User Permission Graph

Lu et al. propose a bottom-up role engineering approach based on matrix decomposition and graph optimization [5]. They use matrix and graph for the representation of user role permission assignments. Then, by using graph optimization, they find optimal role hierarchies which reduce the cost of administration. The contribution is minimizing the number of edges in the role-based graph. Initially, each user permission set is considered as a possible role. Then, the algorithm identifies two roles such that the optimization criteria (the number of edges + the number of roles) is improved. The improvement is made by merge or split operations based on the set relationship between two selected permission sets. This process is performed iteratively until no more improvement is possible. However, the number of iterations needed to reach the most optimal criteria is not clear.

Vaidya and Atluri introduce the notion of matrix decomposition into the Role Mining Problem (RMP) [6]. They decompose the original user-permission matrix into a user-role matrix and a role-permission matrix. The authors also introduce an edge-based role mining problem based on the analysis of the different role mining problems. Meanwhile, they use the matrix

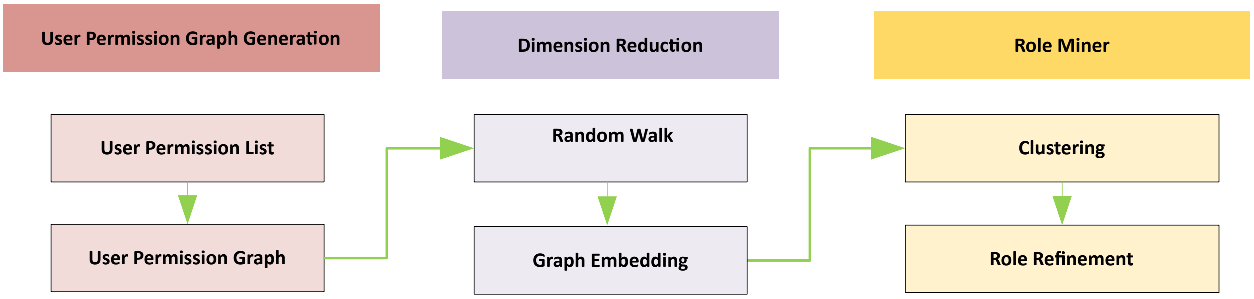

Figure 2. An Overview of the Proposed Methodology

form to denote all kinds of role mining problems. Then, they map the RMP to Minimum Tiling Problem and Discrete Basis Problem by showing that RMP and $\delta - RMP$ is NP-complete problem. This approach has a limitation regarding to the reasonable time complexity.

Ene et al. [7] introduce a heuristic role mining method as a process of finding the minimum number of roles. The algorithm chooses a user u and finds a pair $< U(u); P(u) >$ as a role in each step. $U(u)$ means all users who have all of $u's$ permissions. $P(u)$ denotes a user $u's$ permission set. Then all user permission assignments between $U(u)$ and $P(u)$ are deleted and the remained user permission assignments are allowed to pass next iteration.

In [21], the authors propose the HierarchicalMiner that is a formal concept analysis based approach. The proposed approach tries to address a key problem in role mining techniques which is discovering roles with their semantic meanings. They argue that the roles with semantic meanings may be easy to use and maintain in practice. The authors study the role mining problem in two cases. Once when the user permission relation is the only available information and once with the user-attribute information also available. This paper uses WSC (Weighted Structural Complexity) as an evaluation measure to evaluate the resulted role set.

In [8] Takabi et al. suggest a similarity-based hierarchical role mining approach which generates the reduced concept lattice in the first phase, and then prunes this existing lattice and selects the new RBAC state which has the minimum perturbation in comparison with existing RBAC state. The proposed algorithm is not evaluated by real data.

Jafarian et. al solve the optimal role mining problem by transforming the role mining problem to a constraint satisfaction problem [9]. The transformation allows discovering the optimal RBAC state based on customized optimization metrics. The time complexity of the approach is exponential to the size of users and permissions. This approach is not scalable.

Another category of works include machine learning approaches. Prominent ones are clustering-based methods, where the whole RBAC construction of user-role-permission is generated by the clustering algorithms, and the suggested set of roles are given as an output. For example, Schlegelmilch in [10] proposes a method called ORCA that uses the hierarchical clustering method to investigate the user permissions relation and then combines the ORCA tools to illustrate the clustering results in graphical form. Initially, this algorithm clusters the users who have the largest same permissions together. Then, it merges and updates the permissions clusters according to the maximum number of common users to create a new cluster. ORCA repeats the process until no user has permission in any two clusters. In this method, the permissions in the generated role sets cannot overlap. Since in typical use cases permissions are often shared among roles, this drawback significantly limits the accuracy of the model.

4. Proposed Methodology

In this paper, we aim to address the role mining problem in migration from DAC-based access control to the role-based access control model. Our goal is to present a scalable and accurate approach for extracting roles from a given matrix of user-permission assignments.

4.1. Problem Definition

Definition 1. (User Permission Graph), a user permission graph, is a bipartite graph defined as $G = (V, E)$ where V is the set of vertices and E is the set of edges. V includes two types of vertices, i.e., user-type vertices and permission-type vertices, as illustrated in Figure 1. Each edge $e_{ij} \in E$ connects a user-type vertex (u_i) to a permission-type vertex (p_j) which means that the j_{th} permission is granted to the i_{th} user, according to the given access control matrix. Now, we should extract roles defined as a set of permissions from this large user-permission graph.

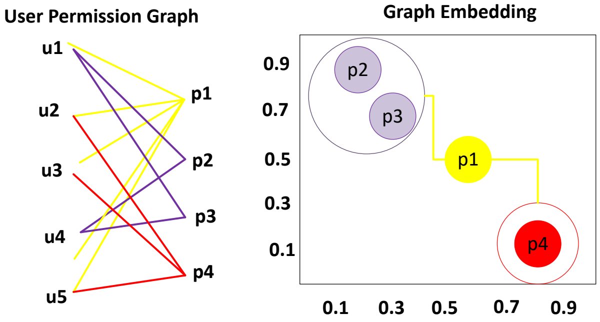

Figure 3. Example of Clustering

4.2. Proposed Procedure

Our role mining approach consists of three steps including graph embedding, clustering and role refinement algorithm as summarized in Figure 2. Algorithm 1 describes these steps in more details.

User-Permission Graph Embedding. At this step first, we convert user-permission assignments to user permission graph. To generate the user permission graph, users and permissions are considered as vertices. In this case, if permission p_j is assigned to the user u_i, new edge (u_i, p_j) is added to the graph. As a result, $G(V, E)$ is generated where V contains users (U) and permissions (P), and E represents user-permission assignments. Second, for efficient processing of user-permission graph, a graph embedding method is used to represent vertices in a low-dimensional space. Graph Embedding methods based on the Skip-gram architecture have been originally developed in the context of natural language [17, 15]. Given the linear nature of the text, the neighborhood of a word can be naturally defined using a sliding window over subsequent words. User permission graphs, however, are not linear and not homogeneous, and thus a different concept of a neighborhood is needed [17].

In our case, random walk is used to capture the shared similarities in local and global graph structure between vertices. Therefore, vertices which have similar neighborhoods in random walks will have similar representations. In other words, permissions that have similar users as their neighborhood will have similar features or representations in the new space. Therefore, the sampling strategy here is to run random walk for each node starting from a permission node and ending in a permission node. In particular, our algorithm works as follows. First, we model a vertex as a function of its node co-occurrences using our random walk sampling. These sequences capture the context around each vertex in the graph and encode neighborhood structures [17]. A traditional way to estimate the likelihood of a vertex v_i co-occurring with its local neighborhood is as follows:

$$Pr(v_i \mid \phi(v_1), \phi(v_2), ..., \phi(v_{i-1})) \qquad (1)$$

However, computing this conditional probability in a large graph is computationally complex. Recently, to address this scalability issue, instead of using the context to predict a missing vertex, vertex is used to predict local structure [15]. The optimization problem to model vertex representations is defined as follows:

$$minimize_{\phi} - \log Pr(v_{i-w}, ..., v_{i+w} \setminus v_i \mid \phi(v_i)) \quad (2)$$

Accordingly, the conditional likelihood of every source-neighborhood node pair is modeled as a softmax unit parameterized by a dot product of their features:

$$Pr(v_i \mid v_j) = \exp\Bigg(\phi(v_i) \cdot \phi(v_j)\Bigg) \sum_{j \in V} \exp\Bigg(\phi(v_i) \cdot \phi(v_j)\Bigg) \qquad (3)$$

Since $\sum_{j \in V} \exp\Bigg(\phi(v_i) \cdot \phi(v_j)\Bigg)$ is expensive to calculate for large graphs, authors in [22] introduce negative sampling. In this case, a very small set of nodes are sampled from the graph for the construction of softmax. Intuitively, optimizing this objective function will make two vertices that are strongly connected in the original graph also close to each other in the embedding space, which preserves the local and global similarity. We optimize the objective function mentioned in Equation 2 using standard back propagation with stochastic gradient descent. We use the default learning rate as 0.025.

As illustrated in Figure 3, Permissions 2 and 3 are close in the embedding space because the context around Permission 2 and 3 are similar in random walk as they are connected to similar users.

Candidate Role Extraction. After representing the user-permission graph in the new low-dimensional space, we employ well-known clustering techniques including k-means [23] and GMM [24, 25] with optimal parameter values to extract candidate roles. To identify optimal parameter values for parameters, we use different initializations to train multiple models and then choose the model with minimum clustering error.

Each resulting cluster corresponds to a candidate role in our process. The respective role is extracted based on only the permission nodes since a role is defined as a set of permissions. In other words, each cluster represents the most similar permissions granted to the most similar users. As shown in Figure 3, the permissions are partitioned in three clusters. Each cluster is considered as an extracted candidate role.

A fundamental issue in clustering is to choose the optimal number of clusters (k). Generally, k depends on the similarity measures and partitioning parameters. In our synthetic dataset, we know the number of original roles in advance, but in a real situation, we do not know the size of the roles beforehand, thus finding the optimal value for k is difficult. In this situation, to estimate the optimal number of clusters, we employ two popular methods, including Elbow method and Silhouette score analysis [26].

The Elbow method seeks to minimize the total within-cluster sum of squares (WSS) as a function of number of clusters intuitively. k will be selected as the number of clusters when adding another cluster does not significantly improve the clustering accuracy [27]. The Silhouette approach determines how well each point lies within its cluster. In other words, it measures the quality of a clustering method by doing an investigation on the separation distance between the output clusters. The indication for a good clustering is a high average Silhouette score [28].

By increasing the value of k, the clustering error will eventually decrease, but the number of extracted roles will also increase, which means more complexity. So, it is important to select k by striking a trade-off between accurate role mining and complexity.

Role Refinement. From the security perspective, it is critical to ensure that migration from DAC to RBAC does not impose security impacts on the organization. In other words, if an assigned extracted role to a user has more permissions than the required permissions for a specific job, then this would result in advere security side effects. On the other hand, the assigned role should not have fewer permissions than required to complete a job, since it leads to failure to perform the assigned duties. Indeed, each user must have the same permissions in the RBAC model as in the original DAC model, which is formally defined by Equation 4. To this aim, in the next section, we propose a refinement step. We cover all corner cases and make a role set such that the built UP matrix by Equation 4 is complete as the input UP matrix.

$$UP_{n\times m} = UR_{n\times r} \times RP_{r\times m} \tag{4}$$

Where n is the number of users, r is the number of roles and m is the number of permissions.

Prune and refine step is a heuristic algorithm that mitigates the possible security impacts. In this step, we work on the candidate extracted roles. Alg. 1 shows the pseudo-code for our approach. First, we find the minimal number of roles with the maximum coverage of permissions. To this aim, the coverage of each extracted role and u's permissions is calculated. Then, the maximum is selected. This process is repeated until any current extracted roles find no more coverage. Then, we find roles in R, which cover the permissions of u exactly. Also, the roles may have more needed permissions for u. Now, we should find the permissions not still covered by any roles and then define the found permissions as a new role.

Algorithm 1 Role Mining Algorithm

```
$U \leftarrow number\, of\, users$;
$P \leftarrow number\, of\, permissions$;
$UP \leftarrow initialize\, with\, user\, to\, permissions$;
**Define User Permission Graph:**
$G \leftarrow Graph(U \cup P, UP)$
**1. DeepWalk($G$);**
**2. Clustering:**
$C \leftarrow$ Cluster(permissionNodes)
**for** each $c_i \in C$ **do**
    $r_i \leftarrow c_i$
**end for**
$R$ is set of all extracted roles $r_i$
**3. Prune and Refine:**
**for** every user $u \in U$ **do**
    $P_u = UP[u]$
    Find minimum set of roles $R' = \{r'_1, ..., r'_x\} \subset R$,
    such that $R'$ maximally covers $P_u$.
    $P_u = P_u - \bigcup_{i=1}^{x} r'_i$
    **if** $P_u \neq \emptyset$ **then**
        $r_{new} = P_u$
        $R = R \cup r_{new}$
    **end if**
**end for**
```

5. Experimental Evaluation

To evaluate the effectiveness of our proposed approach, we conduct an experimental study. The synthesized data is based on a template used in [29]. The template is generated by researchers from Stony Brook University for a RBAC system in a typical university data system.

5.1. Synthetic Data Generation

The synthetic datasets are created using the following procedure as shown in Algorithm 2. First, a set of roles are defined based on the mentioned template. Next, users are created. For each role, a random number of users up to a specified maximum are chosen (DistFunction in Algorithm 2). Then, the user permissions are set according to the roles to which the user has been assigned. Table 1 shows the characteristic of the dataset generated. Three different datasets for different combination of parameters are created. We performed the experiments using these three synthetic datasets.

Table 1. Characteristics of synthetic datasets

Data Set	numUsers	numRoles	numPermissions
Dataset1	1000	18	37
Dataset2	5000	18	37
Dataset3	27027	430	1000

Algorithm 2 Synthetic Data Generation Algorithm

```
$R \leftarrow number\ of\ roles$;
$U \leftarrow number\ of\ users$;
$P \leftarrow number\ of\ permissions$;
$UR \leftarrow initialize\ at\ zero$;
$RP \leftarrow initialize\ according\ to\ the\ template$;
$numberUsersPerRole \leftarrow DistFunction(U, R)$;
for $k \leftarrow 1$ to $R$ do
    $numberUsers \leftarrow numberUsersPerRole[k]$;
    for $i \leftarrow 1$ to $numberUsers$ do
        $user \leftarrow Rand(U)$;
        $UR_{user,k} \leftarrow 1$;
    end for
end for
```

5.2. Experimental Settings

We implemented our role mining algorithm using python and C#. The experiments were performed on a Core-i7 3.60 GHz Intel(R) CPU with 8 gigabytes of memory. We investigated multiple setups, including various number of users and permissions.

5.3. Experimental Result Analysis

Results on small datasets. Based on the experimental results on small datasets (Dataset1 and Dataset2), the proposed RoleMiner extracts an accurate role set in complete accordance with the original dataset. This high accuracy is achieved via Steps 1 and 2 in Algorithm 1. In other words, for small datasets, the combination of graph embedding and clustering work perfectly such that there is no corner cases to cover by prune and refine step.

We select the optimal number of clusters with *Elbow* and *Silhouette* methods. Due to space limitations, we show the elbow method graph for only Dataset2 in Figure 4. As we can see in the figure, for this dataset, $k = 18$ is a good choice of Elbow.

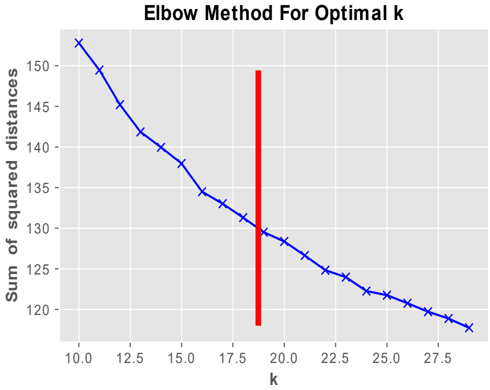

Figure 4. K-means Clustering SSE vs. Number of Clusters for Dataset2

Results on large dataset. To optimize the accuracy of the built model on larger dataset, we need to tune the learning parameters including the number of dimensions in the new embedding space (d), the length of walk (L), and the appropriate number of clusters in the dataset.

Our graph embedding algorithm uses a number of parameters including the length of random walk, the number of walk per node and dimensions of embedded representation. We set d between $(32, 256)$, the length of the random walk between $(32, 256)$, and the number of walks per node between $(10, 100)$. Additionally, the number of negative sampling is set to 5 [22]. In Figure 5 and Figure 6, we examine how the different choices of parameters of graph embedding affect the accuracy. As shown, increasing d (number of features) improves accuracy. Similarly, we also consider how the node neighborhood parameter (walk length) affects accuracy. Figure 6 shows that increasing the length of walk increases accuracy. This is because with longer walks in random walk, more information is collected to define the neighborhood of each node; hence, better representation. It is shown that accuracy saturates when the dimensions of the representations is around 200. Similarly, we observe that accuracy saturates once the walk length is around 100.

We evaluate the performance of the model in terms of accuracy. We use the Jaccard similarity measure to quantify how close the extracted roles are to the original roles [30]. The Jaccard index is a well-known measure used for measuring the similarity and dissimilarity between two sets by computing the ratio of number of common members between the two sets over the total number of members in the union of these two sets. Assuming r is an original role, and r' is the extracted one, the Jaccard similarity measure is defined as follows:

$$J(r, r^{'}) = \frac{(|r \cap r^{'}|)}{(|r \cup r^{'}|)} \quad (5)$$

Accordingly, the accuracy is calculated as follows:

$$\text{Accuracy} = \frac{\text{num of extracted roles with Jaccard value = 1}}{\text{total num of roles}} \quad (6)$$

Table 2 shows the effectiveness of the proposed method given different parameter settings. The changes in dimensionality and walk length while keeping other parameters constant shows that by increasing the dimensionality, accuracy improves. Also, the increase in walk length improves the accuracy of Role Mining algorithm.

Table 2 and 3 represent the impact of increasing the number of clusters while keeping everything else unchanged. By increasing the number of clusters, we see a positive impact on the performance of proposed approach.

The reported performance results are before applying the prune and refine step. This heuristic algorithm refines the extracted roles such that the verified roles resulting at the end of Algorithm 1 are the correct and accurate roles with complete accordance with original role set. In other words, we identify a correct and complete set of roles.

Figure 7 shows the time complexity required to execute graph embedding algorithm for three different data sets. The costliest step in our approach is the graph embedding algorithm. As illustrated, the processing time increases by increasing the walk length for all three data sets.

6. Discussion

The reason we used GMM for clustering is that having shared permissions between two or more roles is typical in RBAC structures, as shown in Figure 3. In other words, we need to use a soft-clustering algorithm. Soft clustering methods assign a score to a data point for each cluster. In our case, it assigns a probability to permission for each role (cluster). The value of the probability indicates the association strength of the permission data to the role cluster. We performed two series of experiments for both small and large datasets. Based on our observations, GMM provides better results for small datasets. For the large dataset, we did not observe a considerable difference in clustering quality between k-means and GMM.

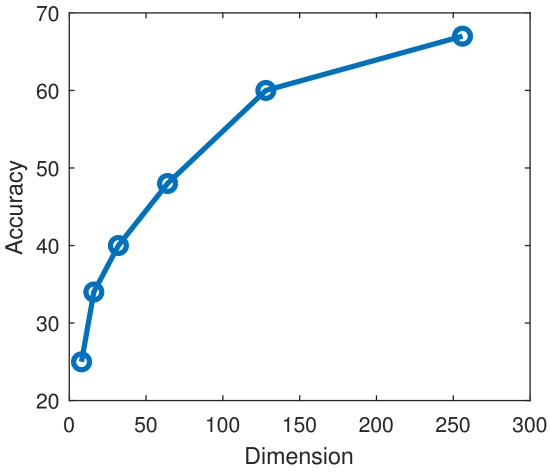

Figure 5. Parameter Sensitivity: Dimension

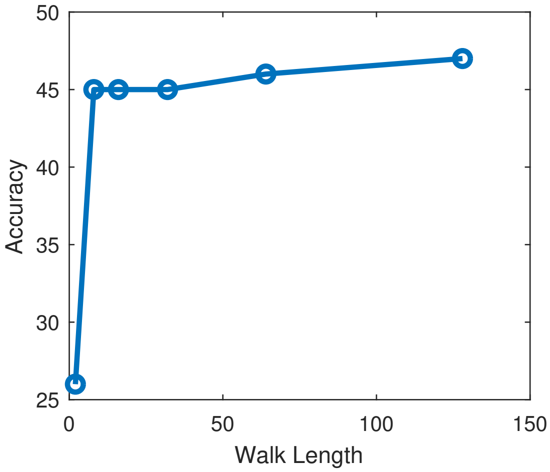

Figure 6. Parameter Sensitivity: Walk Length

In our graph embedding method, we also tried other similarity measures such as direct connection between nodes and common neighbours by considering $W = 1$ and $W = 2$ respectively. However, as shown in the experiment, these similarity measures fail to obtain enough information to learn characteristics of nodes. In those cases, similar permissions did not have similar features in the embedding space. Finally, we selected random walk with longer walk ($W > 2$) to improve the result of clustering.

7. Conclusion and Future Work

In this work, we present an efficient approach to answer the role mining problem in role-based access control models. To this aim, we propose the use of graph embedding and clustering techniques. We model the

Table 2. Role Mining results for the synthetic large dataset

U	R	P	d	L	Clustering Method	numClusters	Accuracy
27027	430	1000	128	32	GMM	300	62
27027	430	1000	256	32	GMM	300	63
27027	430	1000	128	128	GMM	300	59
27027	430	1000	256	128	GMM	300	60

Table 3. Role Mining results for the synthetic large dataset

U	R	P	d	L	Clustering Method	numClusters	Accuracy
27027	430	1000	128	32	k-means/ GMM	400	66
27027	430	1000	256	32	k-means/ GMM	400	67
27027	430	1000	128	128	k-means/ GMM	400	66
27027	430	1000	256	128	k-means/ GMM	400	67

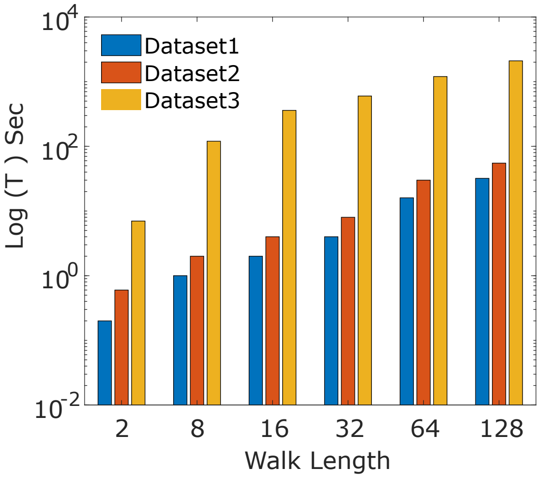

Figure 7. Time vs. Walk Length

problem in graph space and use a random walk sampling based learning algorithm to learn low-dimensional representations for each permission node in the graph. Then, by using the simple clustering algorithms such as k-means and GMM, we cluster the permission nodes as candidate roles. Finally, by employing a heuristic algorithm, the verified, correct, and accurate roles are extracted. The proposed method is efficient in practice, and improves the scalability issue in RBAC implementation in large-scale organizations.

For future work, we expand our approach to consider and extract role hierarchies. As another direction for future work, we will investigate the viability and effectiveness of a multi-step random walk for learning multi-scale relationships between nodes in the embedding space. We will also investigate use of other similarity measures, such as reachability, in efficient learning of node characteristics in a low-dimensional space.

References

[1] B. D. T. H. A. Buecker, J. C. Palacios and I. Yip., “Identity management design guide with ibm tivoli identity,” 2005.

[2] O. A. Ali, K. Najran, T. M. Wahbi, and I. M. Osman, “E-government security models,” *International Journal of Computer Applications Technology and Research*, vol. 5, no. 7, pp. 439 – 442, 2016.

[3] J. Vaidya, V. Atluri, and Q. Guo, “The role mining problem: A formal perspective,” *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 3, pp. 27:1–27:31, 2010.

[4] J. Vaidya, V. Atluri, and J. Warner, “”roleminer: mining roles using subset enumeration”,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pp. 144–153, 2006.

[5] H. Lu, J. Vaidya, and V. Atluri, “Optimal boolean matrix decomposition: Application to role engineering,” in *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, pp. 297–306, 2008.

[6] J. Vaidya, V. Atluri, and Q. Guo, “The role mining problem: finding a minimal descriptive set of roles,” in *12th ACM Symposium on Access Control Models and Technologies, SACMAT 2007, Sophia Antipolis, France, June 20-22, 2007, Proceedings*, pp. 175–184, 2007.

[7] A. Ene, W. G. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan, “Fast exact and heuristic methods for role minimization problems,” in *13th ACM Symposium on Access Control Models and Technologies, SACMAT 2008, Estes Park, CO, USA, June 11-13, 2008, Proceedings*, pp. 1–10, 2008.

[8] H. Takabi and J. B. D. Joshi, “Stateminer: an efficient similarity-based approach for optimal mining of role hierarchy,” in *15th ACM Symposium on Access Control Models and Technologies, SACMAT 2010, Pittsburgh, Pennsylvania, USA, June 9-11, 2010, Proceedings*, pp. 55–64, 2010.

[9] J. H. Jafarian, H. Takabi, H. Touati, E. Hesamifard, and M. Shehab, “Towards a general framework for optimal role mining: A constraint satisfaction approach,” in *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pp. 211–220, 2015.

[10] J. Schlegelmilch and U. Steffens, “Role mining with ORCA,” in *10th ACM Symposium on Access Control Models and Technologies, SACMAT 2005, Stockholm, Sweden, June 1-3, 2005, Proceedings*, pp. 168–176, 2005.

[11] H. Lu, J. Vaidya, and V. Atluri, “An optimization framework for role mining,” *Journal of Computer Security*, vol. 22, no. 1, pp. 1–31, 2014.

[12] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, p. 2319, 2000.

[13] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *SCIENCE*, vol. 290, pp. 2323–2326, 2000.

[14] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS’01, (Cambridge, MA, USA), pp. 585–591, MIT Press, 2001.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’14, (New York, NY, USA), pp. 701–710, ACM, 2014.

[16] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding.,” in *WWW*, ACM, 2015.

[17] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), pp. 855–864, ACM, 2016.

[18] S. Cao, W. Lu, and Q. Xu, “Grarep: Learning graph representations with global structural information,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 891–900, ACM, 2015.

[19] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), pp. 1105–1114, ACM, 2016.

[20] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, “Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM ’18, (New York, NY, USA), pp. 459–467, ACM, 2018.

[21] I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. B. Calo, and J. Lobo, “Mining roles with semantic meanings,” in *13th ACM Symposium on Access Control Models and Technologies, SACMAT 2008, Estes Park, CO, USA, June 11-13, 2008, Proceedings*, pp. 21–30, 2008.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’13, (USA), pp. 3111–3119, Curran Associates Inc., 2013.

[23] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, pp. 100 – 108, 1979.

[24] C. E. Rasmussen, “The infinite gaussian mixture model,” in *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pp. 554–560, 1999.

[25] B. Jiang, J. Pei, Y. Tao, and X. Lin, “Clustering uncertain data based on probability distribution similarity,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 751–763, 2013.

[26] T. M. Kodinariya and D. P. R. Makwana, “Review on determining number of cluster in k-means clustering,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, pp. 90 – 95, 2013.

[27] R. L. Thorndike, “Who belongs in the family?,” *Psychometrika*, vol. 18, p. 267276, 1953.

[28] P. J. ROUSSEEUW, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[29] “RBAC AND ARBAC Policies For a University.” `http://cs.stonybrook.edu/~stoller/ccs2007/university-policy.txt`. Accessed: 6/15/2019.

[30] I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang, and J. Lobo, “Evaluating role mining algorithms,” in *14th ACM Symposium on Access Control Models and Technologies, SACMAT 2009, Stresa, Italy, June 3-5, 2009, Proceedings*, pp. 95–104, 2009.