Preventing Inadvertent Information Disclosures via Automatic Security Policies

Tanya Goyal, Sanket Mehta(✉), and Balaji Vasan Srinivasan

BigData Experience Lab, Adobe Research, Bangalore, India
{tagoyal,samehta,balsrini}@adobe.com

Abstract. Enterprises constantly share and exchange digital documents with sensitive information both within the organization and with external partners/customers. With the increase in digital data sharing, data breaches have also increased significantly resulting in sensitive information being accessed by unintended recipients. To protect documents against such unauthorized access, the documents are assigned a security policy which is a set of users and information about their access permissions on the document. With the surge in the volume of digital documents, manual assignment of security policies is infeasible and error prone calling for an automatic policy assignment. In this paper, we propose an algorithm that analyzes the sensitive information and historic access permissions to identify content-access correspondence via a novel multi-label classifier formulation. The classifier thus modeled is capable of recommending policies/access permissions for any new document. Comparisons with existing approaches in this space shows superior performance with the proposed framework across several evaluation criteria.

Keywords: Digital Rights Management · Extreme Multi-label Learning

1 Introduction

Enterprises constantly share and exchange digital documents containing sensitive information - internally with employees, externally with partners and with customers. With the increase in data sharing, the incidents of data breaches have also increased significantly [2]. A *data breach* is an incident in which sensitive, protected or confidential data, such as personally identifiable information (PII), personal health information (PHI), trade secrets and enterprise financial information is maliciously or inadvertently viewed, stolen or used by unauthorized entities. Until recently, the most common concept of a data breach embodied only malicious attackers hacking into enterprise networks to steal sensitive information. Data Loss Prevention (DLP) e.g. McAfee, Symantec are a class of software that detect and prevent such data breaches by malicious attackers by continuously monitoring sensitive data and providing appropriate encryption based on the sensitivity of the data.

However, according to a study by Johnson [14] in 2008, inadvertent disclosure of sensitive information represents one of the largest classes of security breaches

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 173–185, 2017.
DOI: 10.1007/978-3-319-57454-7_14

exceeding even the number of malicious data attacks. The consequences of such inadvertent data disclosures for enterprises could be severe: compromising customer's privacy, losing market share or damaging intellectual property. DLP software cannot tackle the inadvertent data disclosures since these occur outside the enterprise boundary. This has led to the development of Digital Rights Management (DRM) solutions e.g. Microsoft which are designed to protect sensitive data outside the enterprise boundary as well by ensuring only intended recipients can access the shared sensitive information regardless of their location.

The data being shared in DRM often vary in the degree of sensitive information. A *security policy* is applied to protect it against unauthorized access. A security policy is a collection of information that includes the confidentiality settings and a list of authorized users corresponding to the confidentiality settings. The confidentiality setting in the policy determines how the recipient can use the shared data, for example whether recipients can *print*, *copy*, or *edit text* in a protected document is dictated by the confidentiality settings corresponding to those recipients.

There are two major pitfalls with existing DRM solutions in identifying data breaches. First, the data breach identification in DRM is heavily based on keyword matching with a manually curated dictionary, thus limiting their capabilities severely [13]. Further, the policies in DRMs are manually assigned, which can lead to an error prone process [20]. With the increase in online document transactions, such a manual process is incapable of scaling to the enterprise document volumes, calling for an automated approach to protect against unauthorized access of information.

In this paper, we present an algorithm that analyzes the sensitive information and suggests appropriate access to the documents thus mitigating the risk from inadvertent disclosures. Our algorithm analyzes the historic information and extracts the semantics of the underlying content. The access permissions (that constitutes the document's DRM policy) associated with the information is then analyzed to identify content-access correspondence via a multi-label classifier formulation. The classifier thus modeled is capable of recommending policies/access permissions for any new document.

This paper is organized as follows. In Sect. 2, we describe existing literature in the light of our problem. In Sect. 3, we formulate the DRM policy modeling in a multi-label classification framework [17] and adapt the cost function to simultaneously optimize for precision and recall to suit the needs of the policy modeling. In Sect. 4, we compare the proposed framework against several alternative frameworks along with state-of-the-art security modeling systems showing the viability and superior performances of the proposed framework. Section 5 concludes the paper.

2 Related Work

While automatic recommendation of security/DRM policies for documents is less studied, one direction of explorations that is close to our problem is the prediction of email recipients based on its content. Carvalho and Cohen [8] have proposed an algorithm to generate a ranked list of intended recipients of an email via several algorithmic formulations including an expert search framework and a multi-class framework. Graus et al. [12] address the same problem via a graphical model framework of already composed email messages. Carvalho and Cohen [7] and Liu et al. [16] model the problem of identifying unintended recipients of emails as an outlier detection problem and a binary classification problem respectively based on the textual analysis of the email content overlayed with a correlation based on social network analysis. Zilberman et al. [22] propose an algorithm that extracts topics for all recipients and approves recipients for an email based on its topics and the common topics between the sender and the recipient. All these works focus just on identifying who are the right recipients of an e-mail from a curated list of recipients. However, in the context of DRMs, it is required not only to identify the 'recipients' of a content but also to determine the access permissions of the identified recipients on the content.

Evaluating the *sensitive* nature of information in a content is another direction of exploration that is related to our work. Existing DLP systems use regular expressions [6] and keyword matching to identify PII and other confidential information. Cumby and Ghani [9] present a semi automatic method to identify and redact private content from documents. However, they do not factor in the intended recipients of these documents while making such decisions. Hart et al. [13] propose a binary text classification algorithm to categorize a document as sensitive or non-sensitive. Geng et al. [11] use association mining between different types of PIIs to predict PIIs in emails. All these works are purely based on the document content, without any emphasis on the intended recipients of the content, which is a key factor in our problem.

To the best of our knowledge, there are no prior works that addresses the problem of suggesting appropriate DRM policy by a joint modeling between the document content and its intended recipients (along with their access permissions) which is the novel contribution of our work here.

3 Method

Consider an intra-organizational document repository that comprises of a set of documents with appropriate DRM policies attached to each of them. Let D denote the set of documents in the system, U be the set of all authorized users in the system and ACL be the set of access rights (permissions) in the system. For example, 'read', 'read-modify', 'read-modify-delete' are potential access rights for documents. Each document $d \in D$ is assigned a security policy p which describes the access rights of the authorized users for the document. A security policy is represented as a set of $(user, permission)$ pairs, where each pair defines an authorized $user \in U$ and its corresponding $permission \in ACL$, i.e., $p = \{(user_i, permission_j) | 1 \leq i \leq |U|, 1 \leq j \leq |ACL|\}$. Let P represent the set of all such policies available in the system. Further, we denote L to be the set of all $(user, permission)$ pairs, thus, $|L| = |U| \times |ACL|$.

Given such a repository with documents and their respective security policies, we propose an algorithm to suggest a set of $(user, permission)$ pairs for any new document based on its textual content. More formally, for a new document d, the algorithm aims to provide a ranked list $L^{'} = \left[(user_1, permission_1),\ (user_2, permission_2), \cdots, (user_k, permission_k)\right]$ as the policy to protect the document from unauthorized access.

Feature Extraction: For a given document, we first extract features that capture the personal/sensitive information present in the document. These features include mentions of individuals and organizations, locations, dates, references to currency/money, phone number, social security number and email addresses. We use the Stanford Named Entity Tagger [3] to identify individual and organization names, locations, dates and currency. We further define regular expressions to identify phone numbers, social security numbers and email addresses in the document text. The feature set is further expanded to include individual words found in the document (after removing stop words and stemming the root words) using a "bag of words" representation. TFIDF (Term-Frequency-Inverse-Document-Frequency) [15] is extracted based on the bag-of-words representation for every document d.

With increasing size of document repository, the dimensions of the feature space also increases due to introduction of new words. We therefore reduce the dimensionality by removing features that might be irrelevant to the policy modeling to improve both the overall accuracy and scalability of the model. We calculate the information gain [10] of each feature f in the feature set as,

$$IG(f) = \sum_{l \in L} \sum_{f' \in \{f, \tilde{f}\}} P(f', l).log \frac{P(f', l)}{P(f')P(l)} \tag{1}$$

where, l is a $(user, permission)$ pair. Information Gain measures the amount of information in bits obtained for $(user, permission)$ pair prediction by knowing the presence or absence of a feature, f. We retain the top $k\%$ (70% in our experiments later) informative features for our modeling.

Policy Modeling: Our framework to model security policies aims to suggest $(user, permission)$ pairs for any new document. A brute-force way to formulate this could be as a multi-class classification task where each class represents a security policy. A classifier $g : R^d \longrightarrow P$ can be learned based on the training set $\{(x_i, p_i)|\ 1 \leq i \leq |D|\}$, where $x_i \in R^d$ is the set of features for document d_i, and $p_i \in P$ is the corresponding security policy. For a new document, the above classifier can provide a probability of the existing policies $p_i \in P$ being suitable for the given document which can be used to decide on the final security policy for the document. However, modeling the problem as a multi-class framework problem captures neither the relation between the various security policies nor the overlap between security policies in terms of common users and permissions. Moreover, such a multi-class framework is also incapable of suggesting new policies outside what already exists in the repository and hence can be restrictive.

The above shortcomings can be addressed by formulating the problem in a multi-label classification framework, considering each unique $(user, permission)$ pair as a separate label on the document. In this framework, a function $f : R^d \longrightarrow 2^L$ is learned from the training set $\{(x_i, y_i) \mid 1 \leq i \leq |D|\}$, where x_i is defined as before and $Y = \{y_1,\ y_2, ..., y_{|L|}\}$ denotes the label space. For a new document d_i, the multi-label classifier f predicts the set of labels, i.e., identifies the most relevant $(user, permission)$ pairs that must be assigned to a document as its policy. Unlike the multi-class framework, such a formulation is capable of recommending any set of $(user, permission)$ pairs in the repository.

There are several alternatives for multi-label classification. One class of algorithms transforms the multi-label classification problem to a binary classification, either by training binary classifier for each label in data-set (binary relevance method) [18] or by training binary classifier for multiple label subsets in data-set [21]. Another class of algorithms adapt traditional classification frameworks to the multi-label task [18]. However, both these methods are highly sensitive to the distribution of labels and do not perform well when the labels have very few training examples. Also, as these methods rely on independent models for each label/label sets, prediction cost increases with increasing number of labels.

In light of these drawbacks, a recent exploration in this space is centered on **Fast Extreme Multi-label Learning (FastXML)** which deals with large number of labels having skewed label distributions. FastXML learns a hierarchy over the feature space by recursively partitioning a parent's feature space between its children. The partitioning at each node is done by optimizing a ranking loss function, i.e., normalized Discounted Cumulative Gain (nDCG). Agrawal et al. [5, 17] observe that in the case of XML problems, only a small number of labels are active in each partition of feature space, thus improving the modeling capacity.

However, the ranking-loss function in FastXML [17] only includes a reward for a high recall (correctly predicting all the relevant labels) without accounting for the precision (reducing incorrectly predicted labels) in the prediction. For modeling security policies, it is also important to penalize wrongly predicted $(user, permission)$ pairs, as that means that an ineligible user has been given permission to sensitive information. Given a ranking r of $(user, permission)$ pairs and the ground truth vector y_i, the discounted cumulative gain $L_{DCG}(r, y_i)$ is modified as,

$$L_{DCG@k}(r, y_i) = \sum_{l=1}^{k} \frac{(y_{r_l}) + (y_{r_l} - 1)}{log(1+l)}, \tag{2}$$

where, y_{r_l} is the binary ground truth for the l^{th} label according to ranking r (as defined in [17]), i.e. it has the value 1 if the l^{th} label is attached to document d_i. We add $(y_{r_l} - 1)$ term to the definition of L_{DCG} [17] to introduce a $-1/log(1+l)$ term for each wrongly predicted $(user, permission)$ pair in the top-k labels. This ensures that apart from positive labels predicted with high ranks being rewarded, highly ranked negative labels are also penalized. Algorithm 1 outlines the steps required to obtain the hierarchy.

Algorithm 1. FastXML: Grow Tree

```
Require: $\{x_i, y_i\}_{i=1}^N$
 1: $N_{root} \leftarrow$ new node
 2: if no. of $(user, permission)$ pairs active in $N_{root} < MaxLeaf$ then
 3:     Create $N_{root}$ into a leaf node
 4: else
 5:     Learn linear separator $w$
 6:     $n^+ = \{x_i | w^T x_i > 0\}$
 7:     $n^- = \{x_i | w^T x_i < 0\}$
 8:     $N_{root}$(linear_separator) = $w$
 9:     $N_{root}$(left_child) = GrowTree($\{x_i, y_i\}_{i \in n^+}$)
10:     $N_{root}$(right_child) = GrowTree($\{x_i, y_i\}_{i \in n^-}$)
11: end if
12: return $N_{root}$
```

Finally, the FastXML algorithm learns a linear separator w at each node of the tree, which divides the feature space into the positive and negative partition respectively, by minimizing a ranking loss function given by,

$$min||w||_1 + \sum_i log(1 + \exp(-\delta_i w^T x_i)) - \sum_i (1 + \delta_i) L_{nDCG}(r^+, y_i) \\ - \sum_i (1 - \delta_i) L_{nDCG}(r^-, y_i) \qquad (3)$$

where, $w \in R^d, \delta \in \{-1, +1\}, r^+, r^-$ is the ranked list of labels in the positive and negative partitions respectively. The labels are ranked in decreasing order of the number of documents in the partition that they are assigned to.

Prediction: To suggest $(user, permission)$ pairs for a new document, first a feature representation (x_i) of the new document d_i is extracted as before. The algorithm starts with the root node of each tree in the model and traverses down the tree till it reaches a leaf node. For traversal at each node, it calculates the value of the term $w^T x$ where w is the linear separator at that node. Since the linear separator at each node divides the feature space into two parts, depending on the sign of $w^T x$, the document d_i is passed down to the left child node (if $w^T x < 0$) or the right child node (if $w^T x > 0$) till it reaches a leaf node. Each leaf node contains a subset of points from the training data. The algorithm returns the ranked list of the top k $(user, permission)$ pairs active in the leaf nodes of all trees, where the rank is defined as:

$$r(x) = rank(\frac{1}{T} \sum_{t=1}^{|T|} P_t^{leaf}(x_i)) \qquad (4)$$

where T is the number of trees, $P_t^{leaf}(x_i) \propto \sum_{S_t^{leaf}(x_i)} y_i$ and $S_t^{leaf}(x_i)$ are the label distributions of the set of points in the leaf node that x_i reaches in tree t.

4 Experimentation and Results

The problem of automatic DRM policy assignment can be framed as a multi-class classification problem, with each policy being considered as a separate label. However, we had articulated the shortcomings of such a formulation and overcame these with our multi-label framework. In our experiments, we first compare our formulation with a multi-class framework and show the superior performance of our formulation to test our hypothesis. We also test our algorithm against several multi-label frameworks to show the viability of the proposed algorithm. Finally, we compare the performance of the proposed framework against existing security modeling in the context of email protection.

Dataset: All our experiments were performed on the Wikileaks Cablegate [4] data which includes diplomatic cables sent by the US State Department to its consulates, embassies, and diplomats around the world. Each cable is marked with a group of recipients to whom it was addressed, and a classification scale denoting the security level of the document. The classification scale on each document was one of *Unclassified*, *Limited Official Use*, *Confidential* or *Secret*.

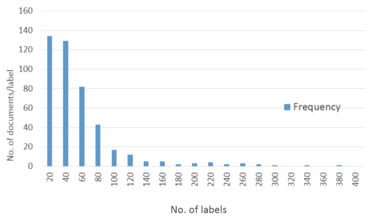
In order to replicate an intra-organization document collection, we considered only the unique documents sent by Department of State for our experiments. Documents with insufficient textual content were discarded. The document's classification scale was used as the access permission given to its corresponding recipients. The set of all such (recipient, classification level) combinations for a document yielded the document's policy for our experiments. The filtered data-set contained 11,760 unique documents, 842 unique policies, 114 unique users/recipients and 452 $(user, permission)$ pairs across all documents. This included 3,301 unclassified, 3,318 limited official use, 3,676 confidential and 1,465 secret documents. Table 1(a) provides additional details about the data-set, at the user level and the $(user, permission)$ level.

Table 1. Wikileaks cablegate dataset

(a) WikiLeaks

Label Description	User	User-Permission
Unique Labels	114	452
avg. labels/doc	5.11	5.11
avg. docs/label	185.01	47.18
min doc/label	49	1
max doc/label	1093	371

(b) $(user, permission)$ distribution.



The average cardinality of labels, i.e., $(user, permission)$ pairs, for documents is 5.11, which indicate that the security policies on each document constitutes a set of 5 $(user, permission)$ pairs on an average. The average number of documents for each $(user, permission)$ pair is 47. However, the $(user, permission)$ pair distribution is highly skewed as seen in Table 1(b) and ranges between a maximum 371 documents with the same user-permission pair to a minimum of 1 document for a user-permission pair.

For all our experiments, the FastXML was built with 50 trees and the maximum number of instances allowed in a leaf node (MaxLeaf) was set at 10. The number of labels k in a leaf node whose probability scores are retained was set to 10.

4.1 Comparison with Multi-class Frameworks

Here, we compare our proposed method against two multi-class formulation to check the viability of our formulation over the multi-class formulation. The first framework is a **frequency based approach**, where all candidate policies are ranked in decreasing order according to the number of times they were assigned to any document in the training set. The top k policies, thus ranked, constitute the set of policies suggested to the user. The second framework is a **1-vs-All approach** where a separate SVM [19] is trained for each policy. For any new document, the scores from each of the corresponding classifiers decides the policy ranking. To obtain a ranking of the policies for our proposed approach, we obtain a binary vector representation ($\{0, 1\}^{|L|}$) of the algorithmically suggested $(user, permission)$ pairs with value 1 for entries corresponding to the top k pairs. Also, we represent all existing security policies present in the system into equivalent binary representations. Then, we use *cosine similarity* metric to identify the nearest security policies and rank them accordingly.

To evaluate the performance of different approaches, we define the $Accuracy@k$ which measures the probability of the actual policy being in the top k predicted policies,

$$Accuracy@k = \frac{1}{|test\_set|} \sum_{doc \in test\_set} \mathbb{1}_{p \in f_k(doc)}, \qquad (5)$$

where p is the actual policy of document doc and $f_k(doc)$ is the set of top-k policies predicted by the algorithm f for the document doc.

Because of the training data requirements of an SVM model, we evaluated only on those policies that had at least n corresponding documents and report the accuracy for various values of n. Note that the proposed approach does not suffer from this limitation since it does not try to generate individual models for each $(user, permission)$ pair.

Table 2 shows the performance of the proposed framework against the multi-class baselines. The results of our experiments indicate the superiority of our proposed multi-label approach in comparison to the multi-class baselines. In particular, the proposed approach performs significantly better in identifying the relevant policy with smaller values of n. For instance, the $Accuracy@10$ for our proposed approach is almost 6% higher than that achieved by 1-vs-All for

Table 2. Comparison against multi-class approaches

Dataset		Freq based		1-vs-All		Proposed approach	
n	#policies	Acc@5	Acc@10	Acc@5	Acc@10	Acc@5	Acc@10
20	95	0.205	0.323	0.565	0.656	**0.581**	**0.682**
30	73	0.226	0.356	0.601	0.691	**0.664**	**0.758**
40	54	0.258	0.406	0.634	0.728	**0.686**	**0.789**
50	43	0.278	0.442	0.685	0.771	**0.713**	**0.791**

the data-set with $n = 30$. This indicates that our algorithm is relatively agnostic to lower number of documents per policy in the training data. As the number of documents per label increases, the performance of the 1-vs-All approach is at par with the proposed approach - indicating that a multi class framework requires a lot of training data per policy to perform on par with the proposed framework.

Real world data-sets often contain numerous policies that might have very few supporting documents to train on as reflected in the Wikileaks data-set as well where only 95 out of 842 policies have been applied to more than 20 documents. Thus, any algorithm that aims to suggest policies by training on such a data-set needs to be robust enough to provide reasonably accurate predictions with lesser documents per policy.

4.2 Comparison with Multi-label Approaches

Next, we compare our approach against multi-label frameworks, 1-vs-All and Rakel [21]. For both these approaches, each $(user, permission)$ pair is considered as a separate label, similar to the proposed algorithm. In the 1-vs-All framework, a separate linear SVM classifier is trained for each unique $(user, permission)$ pair different from the multi-class framework where the modeling was done at the policy level. For Rakel, linear SVM classifiers are built for random ensembles of labels. For any new document, the final score for any label (user-permission pair in our context) is obtained by taking the cumulative score of all ensembles that the label is a part of.

In multi-label frameworks, output is generated at a lower granularity - (user, permission) pairs instead of the entire policy. We therefore use a different set of metrics to evaluate the algorithms in this framework. Since the candidate set of labels is typically large and only a small fraction of those labels is attached to any document, the algorithm needs to be measured on both its ability to correctly predict the highly ranked labels (precision), as well as the ability to retrieve all relevant labels (recall). Here, we define these metrics in the context of our problem. The $Precision@k$ counts the correct pairs in the top k predicted pairs, whereas the $Recall@k$ counts the fraction of the actual pairs that the algorithm is able to predict in the top k pairs. More formally,

$$Precision@K(y', y) = \frac{1}{k} \sum_{i \in rank_k(y')} y_i, \text{ and } Recall@K(y', y) = \frac{\sum_{i \in rank_k(y')} y_i}{\sum_{i \in |y|} y_i}. \tag{6}$$

where y' is the vector of predicted $(user, permission)$ pair vector, y is the ground truth permission set. In our evaluations, we report the average $Precision@k$ and $Recall@k$ across all documents in the test set.

We first evaluate our algorithm's prediction capability at the user level by retaining only the unique users from the k predicted $(user, permission)$ pairs for each document. These are compared against the actual users that have been given some permission by the policy. Modeling at the user level provides a measure of how correctly the set of users who should have permission to a given document is identified. High precision in these cases suggests that fewer irrelevant users are being given access to a certain document whereas high recall suggests that most users that must be given some level of access to the document have been identified. Tables 3 and 4 summarize the results at the user level. Table 4 shows that the algorithm is able to surface 85% of all users if it returns a list of the top 25 labels. Thus, the administrator has to only go through this much condensed list instead of the entire list of 114 users in the system.

Table 3. Precision for different values of k evaluated at user and user-permission level

	Precision@1	*Precision@2*	*Precision@3*	*Precision@4*	*Precision@5*
User level					
1 vs All	0.619	0.557	0.496	0.447	0.408
Rakel	0.508	0.478	0.441	0.4	0.365
Proposed algorithm	**0.653**	**0.609**	**0.548**	**0.502**	**0.463**
User-permission level					
1 vs All	0.34	0.32	0.3	0.286	0.272
Rakel	0.256	0.257	0.259	0.26	0.25
Proposed algorithm	**0.524**	**0.490**	**0.454**	**0.420**	**0.387**

Table 4. Recall for different values of k evaluated at user and user-permission level

	Recall@5	*Recall@10*	*Recall@15*	*Recall@20*	*Recall@25*
User level					
1 vs All	0.502	0.652	0.729	0.767	0.791
Rakel	0.486	0.64	0.674	0.705	0.723
Proposed algorithm	**0.521**	**0.688**	**0.767**	**0.815**	**0.848**
User-permission level					
1 vs All	0.29	0.416	0.483	0.524	0.553
Rakel	0.283	0.42	0.493	0.54	0.57
Proposed algorithm	**0.466**	**0.614**	**0.698**	**0.753**	**0.788**

Next, we evaluate the performance of the algorithms at the user-permission level. As previously elaborated, each policy assigns a unique permission to a user. That is, a policy cannot contain multiple $(user, permission)$ pairs with the same user. However, in multi-label classification, the predicted list of labels may contain multiple labels/$(user, permission)$ pairs corresponding to the same user. We leverage the nature of our problem to solve this. Since the algorithm aims to suggest policies to safeguard against data leak, we hypothesize that a stricter policy is preferable to a more lenient one which may provide extraneous rights to some users. Hence, if a particular user is a part of multiple $(user, permission)$ pairs finally predicted, we make use of the inherent hierarchy in the permissions and assign the strictest permission to that user. For instance, if both (u_i, p_m) and (u_i, p_n) are in the top k predicted labels, we compare the permissions p_m and p_n and retain the stricter permission. In ACL permissions, for example, a $Read$ permission is stricter than a $Modify$ permission, as the latter gives more rights to the user. Tables 3 and 4 shows the results at the $user, permission$ level. Table 4 shows that the algorithm is able to surface 78% of all $user, permission$ pairs if it returns a list of the top 25 labels. This is greatly reduced from the 452 total $user, permission$ pairs that exist in the system.

The proposed algorithm performs the best among all compared frameworks. One reason for this is the ability of our model to handle the skewed distribution of $(user, permissions)$. There exists some $user, permission$ pairs that have been assigned to a very few documents as reflected in the Wikileaks dataset and our proposed algorithm is robust in such scenarios. Traditional methods like 1-vs-All and Rakel, which depend on training independent models for each label/label sets do not perform well in policy modeling with very few training examples.

4.3 Evaluation on Enron Email Dataset

While none of the existing literature addresses the exact problem of assigning security policies for documents, a related exploration is the task of predicting recipients for an email based on its content, e.g. [8]. Due to the robustness of our framework, our approach is capable of handling these tasks as well and here we evaluate the performance of our algorithm for the aforementioned task on the Enron Email Dataset [1] and compare it against the best performing approach in [8]. Our experiments are run on emails sent by 30 Enron users, selected

Table 5. Results on the Enron dataset

	Precision@1	*Precision@2*	*Precision@3*	*Precision@4*	*Precision@5*
Carvalho and Cohen [8]	0.606	0.440	0.352	0.301	0.264
Proposed algorithm	**0.892**	**0.550**	**0.415**	**0.341**	**0.293**
	Recall@1	*Recall@2*	*Recall@3*	*Recall@4*	*Recall@5*
Carvalho and Cohen [8]	0.485	0.640	0.711	0.761	0.789
Proposed algorithm	**0.747**	**0.809**	**0.835**	**0.850**	**0.861**

based on the volume of emails sent, after discarding emails having insufficient textual content. Table 5 provides a summary of the results. The proposed approach clearly outperforms the baseline in [8] thus proving the robustness of the proposed formulation.

5 Conclusion

In this work, we have addressed the problem of automatically recommending appropriate access permissions for users by deciding the appropriate DRM security policies for the document. We have proposed an algorithm that analyzes the sensitive information in the document and determines the right policies for it. Experiments on a real world dataset against several alternate frameworks establish the viability of the proposed approach. Comparison with existing baseline modeling approaches further establishes the superiority of the proposed approach across several evaluation criteria.

References

1. Enron email dataset webpage. http://www.cs.cmu.edu/~enron/
2. The history of data breaches. https://digitalguardian.com/blog/history-data-breaches. Accessed 22 Oct 2016
3. Stanford Named Entity Recognizer (NER). http://nlp.stanford.edu/software/CRF-NER.shtml. Accessed 26 Oct 2016
4. Wikileaks cablegate data. https://wikileaks.org/plusd/
5. Agrawal, R., Gupta, A., Prabhu, Y., Varma, M.: Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 13–24. ACM (2013)
6. Aura, T., Kuhn, T.A., Roe, M.: Scanning electronic documents for personally identifiable information. In: Proceedings of the 5th ACM Workshop on Privacy in Electronic Society, pp. 41–50. ACM (2006)
7. Carvalho, V.R., Cohen, W.W.: Preventing information leaks in email. In: International Conference on Data Mining (SDM). SIAM (2007)
8. Carvalho, V.R., Cohen, W.W.: Ranking users for intelligent message addressing. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 321–333. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78646-7_30
9. Cumby, C.M., Ghani, R.: A machine learning based system for semi-automatically redacting documents. In: IAAI (2011)
10. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4. 5. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 41. ACM (2004)
11. Geng, L., Korba, L., Wang, X., Wang, Y., Liu, H., You, Y.: Using data mining methods to predict personally identifiable information in emails. In: Tang, C., Ling, C.X., Zhou, X., Cercone, N.J., Li, X. (eds.) ADMA 2008. LNCS (LNAI), vol. 5139, pp. 272–281. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88192-6_26

12. Graus, D., Van Dijk, D., Tsagkias, M., Weerkamp, W., De Rijke, M.: Recipient recommendation in enterprises using communication graphs and email content. In: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1079–1082. ACM (2014)
13. Hart, M., Manadhata, P., Johnson, R.: Text classification for data loss prevention. In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 18–37. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22263-4_2
14. Johnson, M.E.: Information risk of inadvertent disclosure: an analysis of file-sharing risk in the financial supply chain. J. Manag. Inf. Syst. **25**(2), 97–124 (2008)
15. Leopold, E., Kindermann, J.: Text categorization with support vector machines. How to represent texts in input space? Mach. Learn. **46**(1–3), 423–444 (2002)
16. Liu, T., Pu, Y., Shi, J., Li, Q., Chen, X.: Towards misdirected email detection for preventing information leakage. In: 2014 IEEE Symposium on Computers and Communications (ISCC), pp. 1–6. IEEE (2014)
17. Prabhu, Y., Varma, M.: FastXML: a fast, accurate and stable tree-classifier for extreme multi-label learning. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 263–272. ACM (2014)
18. Sorower, M.S.: A Literature Survey on Algorithms for Multi-label Learning. Oregon State University, Corvallis (2010)
19. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)
20. Verizon RISK Team: 2014 data breach investigations report (2014)
21. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: an ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Mantaras, R.L., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74958-5_38
22. Zilberman, P., Dolev, S., Katz, G., Elovici, Y., Shabtai, A.: Analyzing group communication for preventing data leakage via email. In: 2011 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 37–41. IEEE (2011)