Auditing Keyword Queries Over Text Documents

Apparreddy Bharath Kumar Reddy[1] * **Sailaja Rajanala**[2] **Manish Singh**[3]
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
{[1]cs15mtech11001, [2]cs15resch11009 }@iith.ac.in,[3]msingh@cse.iith.ac.in

Abstract

Data security and privacy is an issue of growing importance in the healthcare domain. In this paper, we present an auditing system to detect privacy violations for unstructured text documents such as healthcare records. Given a sensitive document, we present an anomaly detection algorithm that can find the top-k suspicious keyword queries that may have accessed the sensitive document. Since unstructured healthcare data, such as medical reports and query logs, are not easily available for public research, in this paper, we show how one can use the publicly available DBLP data to create an equivalent healthcare data and query log, which can then be used for experimental evaluation.

1 Introduction

Large business enterprises, hospitals, etc., maintain a large amount of digital information in the form of structured, semi-structured, and unstructured data. With growing concern among users regarding the privacy of their data, such organizations are required to design a robust data management system. Thus, the goal of DBMS has expanded, to include additional features, such as enforcing data privacy and security (Robling Denning, 1982; Denning et al., 1979), in addition to the primary goal of easy and efficient retrieval.

Lot of research has been done (Duncan and Mukherjee, 1991; Jajodia and Meadows, 1995; Brodsky et al., 2000) to prevent and detect privacy violation for structured data (e.g. SQL) and semi-structured data (e.g. XML) (Byun et al., 2005). However, to the best of our knowledge, there is no existing work that detects privacy violations in access to unstructured text documents using keyword queries. Detecting privacy violations for text documents has not been explored much because it is difficult to audit keyword queries for text data, which is explained later in this section. Organizations tend to maintain their sensitive data in a structured or semi-structured format. However, just as the strength of a chain is equal to its weakest link; similarly, an organization with very secured access to structured and semi-structured can still face privacy violations due to its unsecured unstructured data repositories.

Example *Alice has undergone breast cancer medical treatment in HealthCo Hospital. A few weeks after she returned from the hospital, she started getting advertisements on natural products to treat breast cancer. She blamed HealthCo for disclosing her sensitive disease data to outsiders. HealthCo has a strong security system that will not allow outsiders to directly access Alice's information. HealthCo has to prove that either nobody has misused Alice's private information or find the employees from HealthCo whose access to the information seems suspicious.*

One can use access control policies to secure access to sensitive documents so that only authorized users can access those documents. However, this can restrict access to crucial information at times of emergency. For example, in a hospital, if we create a strict access control policy over medical reports, then it may lead to the inaccessibility of information during crucial hours. An auditing system can help in such scenarios by allowing a relaxed access control policy, and then providing means to detect privacy violations through auditing.

The auditing models that have been proposed for structured or semi-structured data (Agrawal et al., 2004; Bottcher and Steinmetz, 2006; Miklau and Suciu, 2007; Motwani et al., 2008) cannot be used for text documents due to the difference in the query model. Structured and semi-structured data

*Work done during graduate course at Indian Institute of Technology Hyderabad

Proceedings of the 18th International Conference on Natural Language Processing, pages 378–387
Silchar, India. December 16 - 19, 2021.

is often accessed using precise query languages, such as SQL or XQuery, which returns the result using the boolean retrieval model. In these query models, there is no notion of ordering among returned tuples or elements. A query is marked suspicious if its result contains any sensitive tuple, or if a sensitive tuple can be inferred from the query result. The auditing techniques proposed for these types of data do not have a notion of the degree of suspiciousness for a query.

Text documents are commonly accessed using keyword queries, which are not precise. And thus, neither the query nor the result indicates what information the user was looking for. Each query returns a long list of documents ordered by some relevant measure, and in most cases, users may look at only the top few results. The major success of IR is due to the ordered nature of its result set. Thus, rather than just returning a long list of queries that had the particular sensitive document in its result, we need to define a suspiciousness order for the queries, using various factors such as the rank of the document in the result, the relevance of the document to the query, access anomalousness, etc.

Auditing is common in the healthcare domain as it involves sensitive patient data. To our knowledge, there is only one publicly available healthcare dataset[1] of medical reports. The dataset has been reported for around 200 hundred patients and has no associated query log. Since there is no existing big publicly available dataset from healthcare that can be used to evaluate auditing systems for unstructured data. In this paper, one of our main contributions is to model healthcare data using DBLP data, which is a large dataset that contains bibliographic information about computer science journals and proceedings. In the medical domain, access is anomalous if someone accesses sensitive information that one is normally not required to access. Doctors or nurses are allowed to access sensitive information based on their needs. As different staff have different roles in a hospital and the role determines whether an access is anomalous or not, we show how one can model such roles and accesses using bibliographic data.

This paper is organized as follows: Section 2 contains related work. Section 3 discusses the auditing model and system architecture. Section 4 discusses proposed algorithms. Section 6 presents evaluation. Finally, Section 7 concludes the paper with some directions for future work.

[1] https://webscope.sandbox.yahoo.com/catalog.php

2 Related Work

Inspired by the Hippocratic Oath[2], the Hippocratic databases were proposed by (Agrawal et al., 2002) that impose data privacy and security protocols on the data.

(Agrawal et al., 2004) address the problem of detecting privacy violations in the case of relational databases. In this work, the authors proposed a framework for detecting whether or not a relational database is adhering to data disclosure policies. Users specify sensitive information in the form of audit expressions. The audit component takes audit expressions and returns all the queries that accessed sensitive data during execution. (Motwani et al., 2008) also study the problem of auditing SQL queries. Given a forbidden view of a relational database, which should be confidential, and a batch of SQL queries posted over the database. It determines whether a query batch is suspicious or not with respect to the forbidden view. (Stoffel and Studer, 2005) use the database views are used to make the decisions on privacy violations. The work proposes to solve the problem of data privacy by looking for the data leak from a view of the database.

(Bottcher and Steinmetz, 2006) proposed an audit system for sensitive XML databases and XPath query language that uses an audit query to describe sensitive information. It also discusses privacy violations in the case of an attacker who submits multiple queries. (Bertino et al., 2001; Bertino and Ferrari, 2002; Damiani et al., 2000; Kudo and Hada, 2000) propose access control approaches for XML data sources ranging from policies to fragments of XML databases.

Another interesting approach towards data privacy comes from inference methods by (Farkas and Jajodia, 2002). Inference based methods stem from the fact that the access control methods can only block direct access to the sensitive, while there still exists ways of inferring the sensitive data through indirect means. We propose an auditing model for unstructured text documents. Our work is motivated by various works done in the structured and semi-structured database to ensure data privacy.

[2] An oath was historically taken by physicians stating their obligations and proper conduct.

3 Text Auditing System

In this section we walk-through the various components of the text auditing system and further illustrate the working example introduced in Section 1.

Figure 1 presents the skeletal view of the auditing system. Users access the unstructured data repository using keyword queries, where the documents are ranked using any IR ranking algorithm. Given a query, we store the following information in its query log: query ID, query, query timestamp, the ID of the user who issued the query, and the IDs of top-n documents returned for the query. Using the query log we maintain an access index, which greatly improves the performance of our auditing system. Access index is an inverted index from document to queries. For each document ID, we have a posting list that contains the IDs of all the queries that contain the document in their top-n results

A user can submit an audit request $audit(d)$, where d is the sensitive document. The access index is used to find all the queries that had the document d in their top-n result. We call this query set the candidate queries. Here, we assume that users will not be interested in results appearing below the top-n. Henceforth, we first find the candidate set of suspicious queries using the stored query log. A query from the candidate set is termed anomalous if its score crosses a certain threshold. In Section 4, we shall decode how a simple suspicious candidate query transcends into an anomalous one.

Let us carry forward the example in Section 1 in-order to better understand the audit scenario. The Table 1, contains five candidate queries that had Alice's report on breast cancer in their top-n results. For each query, we compute two scores: Suspiciousness score (SScore) and Anomalousness score (AScore). The SScore is a measure of how relevant the query is to the audit document, which also indicates how likely the user has seen the sensitive information. AScore is a measure of how unlikely the query is from the user. The unlikely queries are considered anomalous.

In Table 1, Query 4 has the highest SScore since Alice's blood report contains the information about her breast cancer. Although this query is suspicious, it is not anomalous because Lucy is a Nurse in the Oncology department, and it is normal for her to access such reports. The same argument holds true for Query 1.

Query 2, 3 and 5 are from employees of the Gynaecology and Cardiology department, who are not typically expected to access breast cancer-related information. Although Query 3 and 5 are not asking for any information directly related to Alice, they still have high SScore because Alice's audit document has high relevance for these queries. Both Barbara and Chris might be trying to access the information indirectly. The former is trying to get all breast cancer patients from a particular location, and the latter knows that those who have +ve estrogen receptors are likely to have breast cancer. Although Query 2 is accessing some information about Alice, it has low SScore because it has low relevance to Alice's breast cancer data.

If we want the top-2 anomalous queries, then Query 5 and Query 3 will be returned by our system as they have high suspicious scores and are also anomalous. Given the imprecise nature of keyword queries and the lack of user's background information, such as Role, Department, etc., it is difficult to compute SScore or AScore for queries. In general, it is difficult to get the background information of users as it may not be available or one user may have multiple roles. In this paper, we present an algorithm that computes the AScore without having the prior background information of users who issued the query.

4 Algorithms

In Section 3, we explained how we use the access index to find all queries that had the audit document in the top-n query result. We call them as the candidate suspicious queries. We now discuss the elements that make up the SScore and AScore for each candidate query.

4.1 Suspiciousness Score

The Suspiciousness score (SScore) builds on the relevance of the audit document to a query. A precise query is very likely to pull the relevant document on the top. Such queries will land higher SScores concerning the sensitive document. Next, we study a few popular choices for SScore.

Query Relevance: An IR ranking function returns the documents in decreasing order of similarity score to the query q. We call this similarity score as IRScore.

$$IRScore(q, d) = similarity(R, q, d) \quad (1)$$

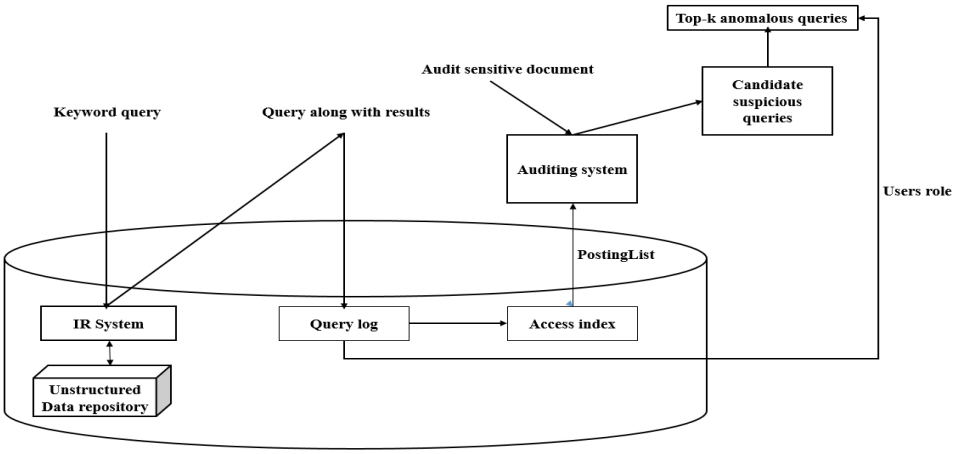

Figure 1: Architecture of text auditing system

Query ID	Username	Department	Role	Keyword query	SScore	AScore
1	Bob	Oncology	Doctor	Non-invasive breast cancer	0.4	No
2	Carol	Gynecology	Doctor	Alice urine report	0.15	Yes
3	Barbara	Gynecology	Nurse	Breast cancer Ann Arbor	0.37	Yes
4	Lucy	Oncology	Nurse	Blood reports of Alice	0.8	No
5	Chris	Cardiology	Doctor	Reports with +ve estrogen receptor	0.45	Yes

Table 1: Sample queries that have accessed Alice's breast cancer report

where R is the IR ranking function, q is the input query and d is the document. If the query has a high similarity with the document, then it indicates that the user who issued the query may be interested in the document.

Document Rank: In IR, results are shown in ranked order of decreasing IRScore i.e., Eq. 1 . The position of a document in the ranked list determines its ease of access. The IRScore can only estimate the significance of a document w.r.t a query. But the relevance of a document is relative to all other documents in the results set. A document catches the attention of the user when it falls within a certain percentile of ranks. In case of a generic query, a majority of the documents have high IRScore. Nevertheless, the user may not look at a document that has high IRScore but does not appear in the top 20-30 documents. On the other hand, indirect queries are vague thus all the documents in the result set have low IRScores. As a result the user can still access the document if it appears in the topmost suggestions (e.g. Query 3 in Table 1) even though it possesses low IRScore.

Therefore, we define IRRank to take into account the rank of result documents in the retrieval system. One can use either the rank of the document or the page number in which the document appears. From empirical evaluation, we observed that considering page number is better than document rank. We thus define IRRank as:

$$IRRank(q, d) = e^{-\lfloor \frac{r}{N} \rfloor} \quad (2)$$

where r is the document rank and N is the number of documents shown per page. By using this function, all the documents that appear on page i have IRRank score of e^{-i}. In general, users only look at the top few pages and the likelihood of their seeing two documents that appear on the same page is equal, so we chose this pagewise exponentially decreasing scoring function.

Click Log: Users do not click on arbitrary links but make selective choices. The click log contains information about the query, the ranked list of documents presented to the user, and the set of links the user clicked (Joachims, 2002). Although a user may not click a sensitive document, by looking at the document snippet in the result set or just its presence in the result set may reveal its sensitive information. Click log information is not definitive of the suspiciousness since it does not record

the cases where the user hovers over the document without clicking on it. As a result, it is not used in our paper.

Time Spent by the User: A user spending lot of time on a particular page indicates that he is interested in the documents present on that page, thus his access should be more suspicious to documents that appear on that page. If we know the amount of the time user spent on each page, we can include it in IRRank by defining it as follows:

$$IRRank(q, d, t_i) = e^{-\lfloor \frac{(\frac{r}{N})}{t_i} \rfloor} \quad (3)$$

where t_i is the time (in minutes) that is spent by the user on a page $\lfloor \frac{r}{N} \rfloor$. The IRRank of all the documents in a page increases if the user spends more time on that result page.

SScore of a query for a given sensitive document d is defined as the product of the IRScore and the IRRank of the query for the given document. The set of Candidate Suspicious Queries (CSQ) can be defined as the top-m queries with respect to SScore, or all the queries with SScore more than some given threshold. We use a threshold in this paper.

4.2 Anomalousness Score

In this section, we compute the anomalousness score for the Candidate Suspicious Queries. We compute the anomalousness score using the following two steps: (a) For each user find the topics of interest; (b) Determine anomalous score for each query that accessed the document by computing how anomalous the query is to the topics of interest of the user who issued the query. We determine the anomalousness of the user's topics of interest by comparing them with the topics of interest of other users who have also accessed the document. We explain these two steps below.

Topics of Interest: To find a user's topics of interest, we consider all the user's queries. We then take the union of top-20 result documents for each user's query and denote it as the set S_u. The user's topics of interest are then computed using three topic modeling algorithms, namely TF-IDF, LDA (Blei et al., 2003) and TNG (Wang et al., 2007), on S_u. The TF-IDF representation is the most straightforward approach computed by selecting the top-k words with the highest TF-IDF score.

LDA is a general probabilistic topic modeling algorithm. It is extensively used to determine important topics and terms from a collection of documents. We apply LDA on S_u to get the user's topic of interest. LDA considers each document as a mixture of topics and places frequently co-occurring terms under the same topic with high probabilities. It computes the document-topic distribution (θ) and term-topic distribution (ϕ), which signify the importance of topics in a document and the importance of terms in a topic respectively. The document-topic distribution (θ) is defined as follows:

$$\frac{N_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} N_{dk}^{DT} + T\alpha} \quad (4)$$

where N_{dj}^{DT} is the number of times a term appears in document d that has been assigned to topic j. D and T stand for the document, topic respectively. α is a smoothing constant. Similarly, term-topic distribution (ϕ) is computed as follows:

$$\frac{N_{ij}^{WT} + \beta}{\sum_{k=1}^{W} N_{kj}^{WT} + T\beta} \quad (5)$$

where N_{ij}^{WT} is the number of occurrences of a word i that has been assigned to topic j. W and T represent the terms, topics respectively. β is a smoothing constant. LDA generates the topics from S_u, and each of these topics contains unigram words (terms).

The above methods do not generate topic phrases. Phrases are important to convey a specific meaning. The meaning of 'natural language processing cannot be completely captured by any of the individual words of this phrase. To overcome this problem, we use TNG, which generates topical collocations as well as better unigram words. We use TNG to generate the topic of interest of users from the document set S_u. Similar to LDA, we generate top-m topics.

Query Anomaly: To determine query anomaly, we take all the queries that had the sensitive document d in their top-n result, say $n = 30$. We use these queries and the query log to find the set of all the users $\mathcal{U}_d$ who had the document d in their top-n result. We say a query is anomalous if the user who issued the query has a topic interest that is very different compared to the document's topic.

However, we cannot directly compute the topics from a document because a document has very limited information. Topic modeling algorithms

generate good topics only if the corpus has a large amount of data. In a small single document, each term would be present only a few times, so we cannot determine the term importance directly from the document. To address this challenge, we do not directly compare the topics of interest of users with the topics in the sensitive document. We use an indirect approach, where we look at all the users who have accessed the document, and from those users, we find users whose topics of interest are anomalous. Our problem can be formally defined as follows:

Problem *Given a set of users* $\mathcal{U}_d = \{X_1, X_2, ...X_m\}$ *who had the document* d *in their top-*n *result. The access anomaly score of user* X_i *is equal to his average distance from his* k*-nearest neighbors in* $\mathcal{U}_d - X_i$.

We use nearest neighbors to define the anomalous score. If a user has topics of interest that are very different from other similar users who have also accessed the document, then that user would get a high average distance score.

Given two users X_i and X_j, we define their similarity as the cosine similarity of their topics of interest. For topics using TF-IDF, we can directly compute the cosine similarity between topics of interest vector by taking each topic as a term and the TF-IDF score as the term importance. However, we cannot use cosine similarity for the topic distribution obtained using probabilistic topic modeling algorithms, such as LDA or TNG. These algorithms will generate a set of topics with document-topic distribution probability (θ), and for each topic, they will generate a set of terms with term-topic distribution (ϕ). The same term may be present in multiple topics with a different degree of term importance. To use cosine similarity we need to have a document-topic vector that has one importance score per term, where the topic could be a unigram or phrase term. To compute this type of vector, for each term we compute its weight by multiplying the document-topic distribution probability (θ) with term-topic distribution (ϕ). The probability θ indicates the importance of the topic and the probability ϕ indicates the importance of the term in that topic. If a term is present in multiple topics, then the score we assign to the term is the maximum value of the product of the corresponding θ and ϕ values.

5 Modeling Healthcare Data using DBLP

This section shows how to create equivalent healthcare data using the DBLP dataset because such healthcare data is not publicly available for research. We first describe our dataset and then explain the modeling of roles and specializations, generate query logs, and find ground-truth anomalous queries. DBLP[3] is a bibliographic dataset containing information of 3.66 million publications from Computer Science. Each publication in the dataset has information such as title, conference name, the name of authors, publication year, etc. We removed publications that do not contain the name of authors, abstract, or conference name from the dataset. Our processed dataset contained 183232 publications.

5.1 Modeling Specializations and Roles

In healthcare, we consider access anomalous if an employee accesses sensitive information that he usually is not required to see. Since in hospital, each employee has his specialization(s) and the department he belongs to, we can compare the accesses of the particular employee with other employees with similar roles and departments to determine whether the access is anomalous or not.

Although this information is not directly available in DBLP data, we use authors' publications to find their area of research. If an author publishes or looks for a very different paper from his research area, then we consider such papers anomalous. For example, Prof. H. V. Jagadish[4] is a well-known researcher in the area of Databases and Data Mining. However, one of his papers: "Hui Jin, H. V. Jagadish: Indexing Hidden Markov Models for Music Retrieval. ISMIR 2002" seems like an outlier given most of his other papers are in Databases.

Since DBLP does not have the research interest of authors, we combine it with WikiCFP[5] to get the research interest of authors. WikiCFP is a website that advertises calls for papers of international workshops, conferences, and journals. While posting CFP for a conference, one can tag the conference with one or more research areas. We crawled this information from WikiCFP to get the research interest of authors.

For each author, we consider all his publications in the DBLP dataset. We use the conference name

[3] https://www.aminer.cn/dblp_citation
[4] https://web.eecs.umich.edu/ jag/
[5] http://www.wikicfp.com/cfp/

Research Area 1	Research Area 2	Jaccard Similarity
Health Informatics	E-health	1
Formal Methods	Verification	0.67
Parallel Computing	HPC	0.41
Education	E-learning	0.36
Machine learning	Verification	0.05

Table 2: Jaccard similarity between research areas

Conference Name	Areas of the conference
VLDB	Databases
ICVS	Computer Vision, **Pattern Recognition**, **Image Processing**
CIKM	Web, Information Management, Data Mining, Information Retrieval, Text Mining, Databases, **Knowledge Engineering**, **Knowledge Management**, **Database**
KDD	Web, Data Mining, Machine Learning, Information Retrieval, Databases, Knowledge Discovery, Data Science, Big Data, **Knowledge Engineering**

Table 3: A sample of conferences and their research areas as crawled from WikiCFP. Highlighted in bold are areas extended using Jaccard similarity.

in the WikiCFP dataset to get the area(s) of the particular publication. Since there are thousands of category labels in WikiCFP, the CFP poster may not label a conference with all the possible labels. For example, the conference International Conference on Computer Vision Systems (ICVS) is only labeled Computer Vision in WikiCFP. But we know that it also belongs to Pattern Recognition, Image Processing, etc. Using the few area labels of conferences given by the CFP poster, we use Jaccard similarity between areas to find all the related labels of the conferences.

Table 2 shows the Jaccard similarity between a few research areas. Jaccard similarity between Health Informatics and E-health is 1, which indicates that these two areas are almost the same. Jaccard's similarity between Machine learning and Verification is 0.05, which indicates that these two areas are very different. Suppose L_c is the set of the labeled area(s) of a conference c such that $|L_c| > 1$. R_c is the set of related areas with Jaccard similarity greater than a threshold $th = 0.25$. We use the extended set of areas $L_c \cup R_c$ to label conference c. Table 3 shows the conferences and their extended areas after using Jaccard similarity. The related areas obtained using Jaccard similarity are highlighted in bold.

5.2 Generating Query and Access Log

We consider the keywords in titles as keyword queries and all the abstracts as the repository of sensitive documents. We consider the authors of the publications as the users who issued queries. A publication is anomalous if the conference area of that publication has very low similarity with the author's overall publication profile. To generate the IRScore and IRRank of queries, we used Apache Solr[6], which is an open-source IR system.

5.3 Finding Ground-truth Anomalous Queries

To evaluate the algorithms discussed in Section 4, we need ground-truth anomalous queries. Given the huge size of the DBLP dataset, it is difficult to manually label all the anomalous queries. In this section, we present a heuristic to generate ground truth about the queries. For this, we use the manually provided category labels, in other words, research areas of conferences to compute profiles of users and documents in terms of category labels.

Given a paper's abstract and its conference, we get the labeled areas of the conference L_c from WikiCFP and then compute the closely related areas R_c, as described in Section 5.1. We create a profile vector of the document by considering the

[6]http://lucene.apache.org/solr/

extended set of areas, where each feature of the vector is a topic area and the feature weight is one. To compute the profile vector of a user, we add up the profile vectors of all his publications, as described above. We then compute the cosine similarity between the publication and user profile vector. If the cosine similarity is below a certain threshold, we consider the particular publication as anomalous.

6 Evaluation

As introduced in Section 5, we carry out our experiments on a surrogate DBLP dataset. For this, we use the ground-truth generated in Section 5.3. The first step in detecting anomalous queries is to find Candidate Suspicious Queries (CSQ). In this regard, Figure 2 shows the relationship between the number of outliers vs SSsore. From the graph one can observe that queries with SScore less than 0.1 are not a threat as there are no outliers with SScore less than 0.1. Thus while finding CSQs, we can exclude all the queries with SScore less than 0.1. Interestingly, queries with high SScore are not a threat either. These queries seem to be issued by genuine users. This is expected for genuine users, as it is okay to access sensitive documents that are of relevance to them. Most of the outliers are concentrated around SScore value 0.2. These queries are either issued by users who are unable to form a proper query, maybe due to a lack of domain knowledge, or those who are trying to make indirect access. Since these queries are less precise, they have low SScore.

Evaluation Metric: In general outlier detection is evaluated using either the precision-recall graph or the ROC curve (Aggarwal, 2015). ROC is a plot of True Positive Rate (TPR) vs False Positive Rate (FPR). ROC has the advantage of being monotonic and more easily interpretable in terms of its lift characteristics in comparison to the precision-recall curve. ROC studies the trade-off between average-TPR and average-FPR. TPR is the number of outliers that were rightly identified while FPR measures how many non-outliers were wrongly classified as outliers. Ideally, we prefer a model that predicts all the outliers (high TPR) while being specific of not wrongly predicting normal data as outlier (low FPR). A perfect ROC curve would require the curve to stick to the left-hand side; maintaining high TPR and low FPR, and is thus said to have a high area under the curve (AUC).

Topic modeling evaluation: As discussed in Section 4.2, user representations can be computed using three topic modeling algorithms, namely TF-IDF, LDA, and TNG. From the ROC plot shown in Figure 3, we observe that TF-IDF and LDA closely follow each other. Similar to TF-IDF, LDA depends on the frequency of a word to assign a topic to it. Since abstracts are very small texts both TF-IDF and LDA have comparable outcomes here. However, TNG identifies more topic-specific terms by learning n-grams in the topics and can make a clear distinction given the small text. This is also evident from the high AUC under TNG. For the remaining evaluation, we use TNG to get the user representations.

Effect of number of neighbors: The next parameter we evaluate is the optimal number of neighbors k in computing access anomaly. The value of k depends on the nature of the data. For example in a hospital, not every department has an equal number of staff. Suppose the hospital specializes in cardiac treatment and thus has a huge cardiology department, in comparison the ophthalmology department is tiny. If we are looking for staff in the ophthalmology department, we can only get a few similar staff. If we put a big value of k, then our anomaly detection will not be accurate as we will include neighbors who are very different from the staff. The k value changes depending on the data distribution (Latourrette, 2000), it is therefore necessary to understand the data dynamics.

To find the optimal k for different data dynamics, we divided our data into three classes based on outlier density. Figures 4, 5 and 6 shows the effect of k on low, medium and high outlier density. We can observe that while $k = 2$ is an insufficient number of neighbors, $k = 15$ is too big a number to still be called 'nearest neighbors'. Both of them are bad estimators with very low AUC. Plots for all k values except $k = 6$ have close performance. For all three cases, $k = 6$ seems to give the optimal classification. For low outlier density, the number of outliers is less, therefore ROC plots are oriented towards the y-axis. However, for medium and high outlier density, the ROC curves gravitate away from the y-axis as they have a high number of outliers. All these observations are perfectly captured in Figure 7, which is the consolidated ROC plot.

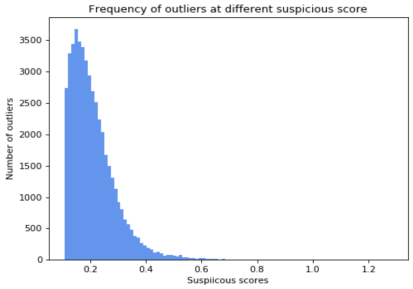

Figure 2: Outlier frequencies at varying Suspiciousness Scores

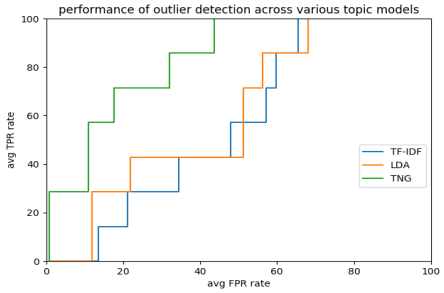

Figure 3: ROC plot for various topic models

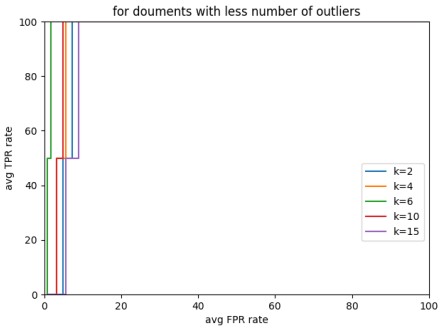

Figure 4: ROC for low outlier density

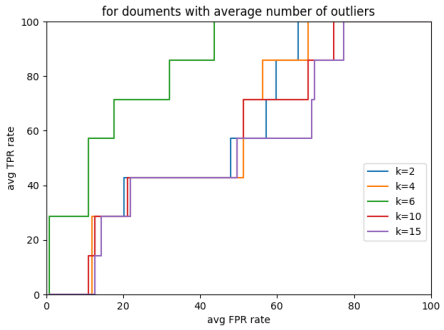

Figure 5: ROC for medium outlier density

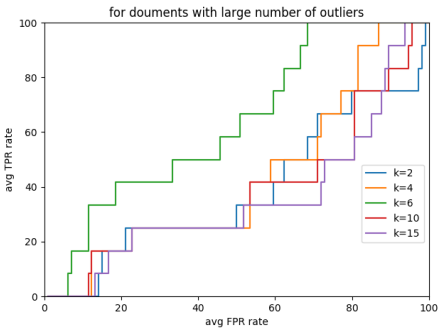

Figure 6: ROC for high outlier density

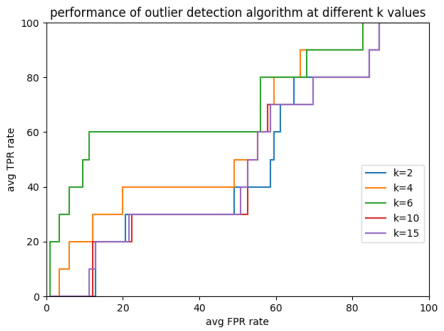

Figure 7: Overall ROC

7 Conclusion

We present one of the first of its type approaches to detect privacy violation in access of unstructured text documents using keyword queries that is mainly useful for healthcare domain. Since healthcare data is difficult to obtain, we also demonstrate the construction of a substitute dataset for healthcare. The proposed system shows promising results.

References

Charu C Aggarwal. 2015. Outlier analysis. In *Data mining*, pages 237–263. Springer.

Rakesh Agrawal, Roberto Bayardo, Christos Faloutsos, Jerry Kiernan, Ralf Rantzau, and Ramakrishnan Srikant. 2004. Auditing compliance with a hippocratic database. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 516–527. VLDB Endowment.

Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. 2002. Hippocratic databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 143–154. VLDB Endowment.

Elisa Bertino, Silvana Castano, and Elena Ferrari. 2001. On specifying security policies for web documents with an xml-based language. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 57–65. ACM.

Elisa Bertino and Elena Ferrari. 2002. Secure and selective dissemination of xml documents. *ACM*

Transactions on Information and System Security (TISSEC), 5(3):290–331.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Stefan Bottcher and Rita Steinmetz. 2006. Finding the leak: A privacy audit system for sensitive xml databases. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 100–100. IEEE.

Alexander Brodsky, Csilla Farkas, and Sushil Jajodia. 2000. Secure databases: Constraints, inference channels, and monitoring disclosures. *IEEE Transactions on Knowledge and Data Engineering*, 12(6):900–919.

Ji-Won Byun, Elisa Bertino, and Ninghui Li. 2005. Purpose based access control of complex data for privacy protection. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110. ACM.

Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. 2000. Securing xml documents. In *International Conference on Extending Database Technology*, pages 121–135. Springer.

Dorothy E Denning, Peter J Denning, and Mayer D Schwartz. 1979. The tracker: A threat to statistical database security. *ACM Transactions on Database Systems (TODS)*, 4(1):76–96.

George T Duncan and Sumitra Mukherjee. 1991. Microdata disclosure limitation in statistical databases: Query sizeand random sample query control. In *null*, page 278. IEEE.

Csilla Farkas and Sushil Jajodia. 2002. The inference problem: a survey. *ACM SIGKDD Explorations Newsletter*, 4(2):6–11.

Sushil Jajodia and Catherine Meadows. 1995. Inference problems in multilevel secure database management systems. *Information Security: An integrated collection of essays*, 1:570–584.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

Michiharu Kudo and Satoshi Hada. 2000. Xml document security based on provisional authorization. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 87–96. ACM.

Mathieu Latourrette. 2000. Toward an explanatory similarity measure for nearest-neighbor classification. In *European Conference on Machine Learning*, pages 238–245. Springer.

Gerome Miklau and Dan Suciu. 2007. A formal analysis of information disclosure in data exchange. *Journal of Computer and System Sciences*, 73(3):507–534.

Rajeev Motwani, Shubha U Nabar, and Dilys Thomas. 2008. Auditing sql queries. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 287–296. IEEE.

Dorothy Elizabeth Robling Denning. 1982. *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc.

Kilian Stoffel and Thomas Studer. 2005. Provable data privacy. In *International Conference on Database and Expert Systems Applications*, pages 324–332. Springer.

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *icdm*, pages 697–702. IEEE.