Misuse Detection for Information Retrieval Systems*

Rebecca Cathey, Ling Ma, Nazli Goharian, and David Grossman
Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
{cathey, maling, goharian, grossman}@ir.iit.edu

ABSTRACT

We present a novel approach to detect misuse within an information retrieval system by gathering and maintaining knowledge of the behavior of the user rather than anticipating attacks by unknown assailants. Our approach is based on building and maintaining a profile of the behavior of the system user through tracking, or monitoring of user activity within the information retrieval system. Any new activity of the user is compared to the user profile to detect a potential misuse for the authorized user. We propose four different methods to detect misuse in information retrieval systems. Our experimental results on 2 GB collection favorably demonstrate the validity of our approach.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval;
H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*User Profiles and Alerts*;
H.4 [**Information Systems**]: Information Systems Applications

General Terms

Algorithms, Experimentation, Security

Keywords

Information retrieval, misuse detection, user profile, relevance feedback, clustering

1. INTRODUCTION

The most valuable resource we have today is information and as such requires appropriate management and protection against misuse and intrusion. Intrusion is generally performed by people who are unauthorized, or outside of an organization, and wish to remain unidentified. The results of intrusions may be catastrophic and therefore a great deal of development has been done in the intrusion detection and prevention area. However, we are confronted with the situation that an authorized user, who can access an information retrieval (IR) system, might misuse the authorization and retrieve and view documents that are available in the document collection but should not be viewed by that user. An insider is someone who has authorized access to the system; while an intruder is someone who does not have authorized access to the system. Our focus in this paper is primarily on detection of the misuse of insiders in an information retrieval system. One way to prevent such situations is to create a user profile that contains some direction toward the kind of documents the user should be able to view. By comparing items to the user's profile, misuse can be detected in a system. Misuse detection systems offer a cost-effective compromise to establishing and assuring a certain degree of security in a system [2].

In this paper we propose four different methods for detecting insider misuse of information retrieval systems. Section 2 gives prior work in misuse and intrusion detection, section 3 gives an overview of the four methods we propose for misuse detection, section 4 give our experimental frame work and results, and finally we conclude in section 5 and give future directions for this work.

2. RELATED WORK

Previous work done on detection can be divided into intrusion detection and misuse detection. Intrusion detection deals mainly with attack to the system from outside. Misuse detection deals mainly with the attack to the system by an authorized user who is misusing his/her privileges.

Some previous work on intrusion detection has been in the area of pattern matching [6] [13]. Also [8] proposes a method for intrusion detection based on text clustering. Text clustering is used to classify program behavior rather than user behavior. The processes are clustered and the system calls from processes are compared for intrusion detection.

Misuse detection has generally been employed to complement the shortcomings of other prevention techniques [2]. Prior work on misuse detection has been mainly focused on using logs and user profiles. Profile-based detection systems audit the deviation of user activities from normal user profiles. In the past, a user's command history has been reviewed based on the percentage of commands used over a specific period of time [10]. The logs are then scanned and

*This work is supported in part by the National Science Foundation under contract # 0119469.

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

mined [9] for interesting temporal and sequential patterns about a user's activity [14].

A successful misuse detection system must overcome many challenges. First of all, a user's profile may change over time. To handle dynamic profiles, learning algorithms are required to track user behavior and adapt to a dynamically changing concept [7].

Chung et al. in [2] describe their misuse detection system, DEMIDS, for database applications. DEMIDS uses the access information of the user to the database tables, columns, and other structures to build a user profile to track the behavior of the user.

In the specific area of information retrieval systems, however, we are unaware of published work directly relating to user query profile learning and abnormal query behavior detection. We implement a misuse detection warning by comparing user behavior to user profile, learned through clustering, relevance feedback, and fusion methods, thus creating a new dimension to profile-based misuse detection for information retrieval systems. This work is based on the initial work disclosed in [3]. In addition, we improved the accuracy of the results and evaluated the effectiveness of our system.

3. METHODS FOR MISUSE DETECTION OF IR SYSTEMS

We proposed and implemented a set of algorithms to build a user profile and detect anomalies in user behavior. A potential misuse of the information retrieval (IR) system can be indicated by comparing a user's actions against his/her profile. Each algorithm may independently flag certain anomalies. Together, the algorithms may be used to increase the likelihood of detecting a misuse. The algorithms are based on some known techniques used in information retrieval system, namely, clustering and relevance feedback. The readers are referred to [4] [12] for general reading on information retrieval related material.

The proposed process of detecting misuse in an information retrieval system is a process that has two distinct stages, which will be described in detail below.

Step 1: Build the Profile

The first step in detecting misuse is to build a user profile. During this step, profiles are built and stabilized for each user based on the user's queries over a period of time. A general assumption is made that the user will only ask queries that are not considered to be a misuse of the IR system. We understand that a sophisticated user, who tends to misuse the IR system, may plan queries in such a way that the profile is not a valid. However, this is an issue that is not addressed in this paper.

Step 2: Test Profile

In the second step, we test the profile to determine the level of misuse by generating a degree of warning. The user's new queries to the IR system are tested against the user's profile. A misuse warning is then computed by comparing the difference between the new queries and the user profile based on each proposed method.

Next, we describe each of the four methods proposed for detecting misuse of an authorized user. These methods are based on a) clustering documents, b) clustering query results, c) relevance feedback, and d) a fusion of these methods.

3.1 Method 1: Clustering Documents

Document clustering groups documents by content and thus reduces the search space required to respond to a query [4]. There are several sequential and parallel clustering algorithms [15] [5] [1]. We use the parallel buckshot clustering algorithm because it provides high-quality clusters in an acceptable time of $O(\frac{kn}{p})$, where k is the number of clusters; n is the number of documents; and p is the number of processors. Before the documents can be grouped into clusters, the documents must first be parsed into their respective terms. We parsed the documents into vectors according to the vector space model, which is a model used to represent the documents and queries in information retrieval systems [4] [12]. A similarity measurement technique is then applied to determine how similar the documents are by determining how their respective vector representations are similar. Similar documents are then grouped into one cluster. As the result of this process k clusters are created. We used the *complete linkage* similarity measure to calculate the similarity between the documents. This technique takes the minimum similarity between documents [4].

Our motivation behind using clustering for misuse detection for information retrieval systems is that all documents in each cluster are similar. Thus, if a user generally queries documents within a cluster, that user should also be able to retrieve any document in that cluster. Therefore, the user's profile is built with clusters deemed to be valid for that given user to query from. The process starts with creating a clustered collection. During the phase of building user profile, the documents retrieved from a user's query are associated with clusters that are then added to the user's profile. This process continues for each user over a period of time until a user's profile is stabilized. As a document that appears at the beginning of the set of returned results has a higher relevancy than a document that is returned near the end of the result set, a user's profile can be stabilized by only adding the associated clusters for the top m result documents to the profile. Once the profile is built, any subsequent query is tested against it. If a user's query retrieved results are far from any of the profile clusters, then a higher level of misuse warning occurs. The algorithm for detecting misuse in an information retrieval system using this method is shown in Algorithm 1.

ALGORITHM 1. *MISUSE_CLUSTERING_DOCS*

```
1. Cluster all documents in collection
2. Build user profile
   profile ← null
   For each query
      ∀ d ∈ resultSet
         identify cluster such that d ∈ cluster
         if cluster ∉ profile
            profile ← profile ∪ cluster
   Continue until profile is stabilized
3. Test user profile
   For each query
      warning ← 0
         ∀ d ∈ resultSet
            identify cluster such that d ∈ cluster
            if cluster ∉ profile
               warning← warning + warningLevel(d)
      Output warning
```

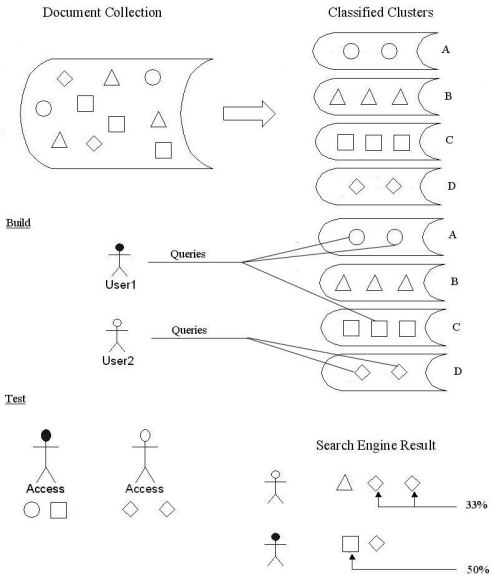

Figure 1: Process of Clustering Documents method

In Figure 1 the sample document collection is classified into four separate clusters. In the building phase, user1 submits queries that return documents from the clusters A and C, whereas user2 submits queries that return documents only from cluster D. Therefore, user1's profile allows access to documents from cluster A and C while user2's profile allows access to documents from cluster D. Then in the testing phase, user1 submits a query that returns two documents, one from C and one from D, while user2 submits a query that returns one document from cluster B and two documents from cluster D. The misuse warning for both users is computed from the number and order of the allowed documents returned, as described in section 3.5.

3.2 Method 2: Clustering Query Results

Our next misuse algorithm uses result set clustering instead of clustering the entire document collection. This method creates profiles based on the result sets of user queries over time. We used hierarchal clustering for this method because it consistently yields the same division of clusters. Although hierarchical clustering is $O(n^2)$, the number of documents that needed to be clustered was much less than our previous method. During the building step, a user submits queries and a set of clusters are built from the retrieved results. The user's profile is stabilized by removing any outliers, or documents that do not match up with any clusters of retrieved results.

In the testing step, any user query is tested against the user's profile. If the similarity of the retrieved documents to the user's profile (i.e. to the centroid of the clusters in user profile) is larger than a defined threshold then a misuse is detected. A similarity measurement between each retrieved document and the cluster centroids in the user profile is computed. If this exceeds a threshold, a potential incident of issue may be the cause. Naturally, it is possible a user is issuing a query that brings back diverse documents. If warnings are continually generated then the system should be alerted. The algorithm for this process is shown in Algorithm 2 and illustrated in Figure 2. As the diagram illustrates, the user's profile consists of clusters built from the documents returned by the user's queries. In this example the documents are classified into three clusters, the triangular, square, and circular documents. Then in the testing phase, the user's query returns a square and an oval document. The square document already belongs to the cluster B, however, the oval document is not a member of any of the clusters. The distance is measured between each of the document clusters and the oval document to determine which of the clusters the oval document is closest to. Since the oval document is closest to cluster C, that distance is used when calculating the misuse warning of this query. The definition for the misuse warning is given in section 3.5.

ALGORITHM 2. *MISUSE_CLUSTERING_RESULTS*

1. *Build user profile*
   documents $\leftarrow$ *null*
   For each query
      $\forall$ $d \in resultSet$
         $documents \leftarrow documents \cup document$
   Cluster documents
   Continue until profile is stabilized
2. *Test user profile*
   For each query
      $warning \leftarrow 0$
      $\forall$ $d \in resultSet$
         For each $c \in clusters$
            $distance \leftarrow max(distance(c, d))$
         update warning
      Output warning

3.3 Method 3: Relevance Feedback

Relevance feedback is a very common technique [4] in IR systems that have been repeatedly shown to improve the effectiveness of a search. The essential idea is that a user query may be expanded to include terms found in an initial set of documents. The process is that a user submits a query and manually identifies terms that occur in the top

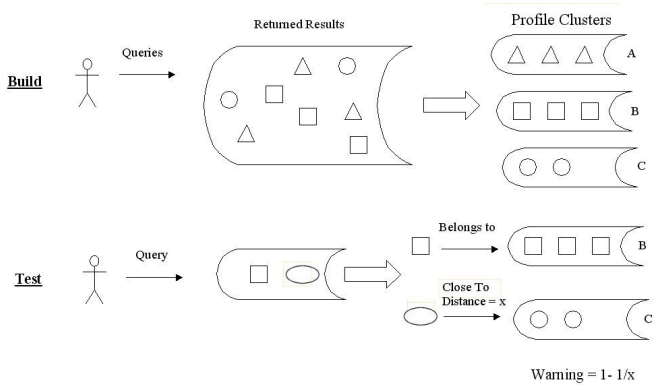

Figure 2: Clustering Query Results Process

documents to add to the query. Pseudorelevance feedback eliminates the manual step and uses term weights to automatically select potentially "good" terms to add to the query. During this process, the top m retrieved terms out of top n retrieved documents are used to modify the initial query.

Instead of constructing user profiles based on query results, we build them based on the query terms. In the building step, a user submits a query. The results for the query terms are then returned and relevance feedback is applied to find "good" terms from the relevant documents. The terms identified by relevance feedback and the original query terms are then added to the user's profile. A profile may be stabilized by altering the values of m and n. Once the profile is built, any user query is tested against the user's profile. An occurrence of misuse is detected if any of the user's query terms are absent from the user's profile that contains the valid user's query terms. The algorithm for this method is shown in Algorithm 3 and is illustrated in Figure 3. As the diagram depicts, during the building phase, the user submits a query consisting of two words: "English" and "Channel". Relevance Feedback returns one term "chunnel", which is then added to the original terms that together create the user profile. Then in the testing phase, the query terms "English" and "Channel" are both matched to terms in the profile, however, "distance" is not. Thus, the misuse warning for this query is calculated from the total number of query terms and the number of those terms missing from the profile. The misuse warning is described in further detail in section 3.5.

ALGORITHM 3. *MISUSE_RELEVANCE_FEEDBACK*

1. *Build user profile*
   $profile \leftarrow null$
   For each query
      $profile \leftarrow queryTerms \cup RF(query)$
   Continue until profile is stabilized
2. *Test user profile*
   For each query
      $warning \leftarrow 0$
      $\forall\ terms \in query$
         If $term \notin profile$
            $warning \leftarrow warning + warningLevel(term)$
      Output warning

3.4 Method 4: Fusion method

Our final method for detecting misuse for information retrieval systems is a combination of the previous three methods. The previous methods are combined by taking the average of all the warnings with the ability to assign each method different weights. By assigning weights, this method can be tailored to fit specific applications.

In the building phase, a user will submit a query. The query results are used to build three profiles, one for each previous method. Once the profiles are built, any query a user submits is tested against all three profiles and an average misuse warning is returned from each individual misuse detection algorithm.

Note that this method can be the same as any other method by setting the weights for the other two methods equal to zero.

3.5 Measuring Misuse Warning

To identify a misuse from the behavior of the user activity, we defined a misuse warning measurement. The misuse warning is a number between 0 and 1 that indicates a degree of dissimilarity of the retrieved results or terms from the user profile. The closer the value is to 1, the more dissimilar the retrieved results are to the user profile; the closer the value to 0, the more similar are the results to the profile. We measured the misuse warning in several ways. The easiest is to take the number of terms or results that are absent from the profile and divide that by the total size of the profile. More formally stated:

Definition 1.

$$w = \frac{A}{s}$$

Where $A = \mid docs \in resultSet \wedge docs \notin profile \mid$
and $s = \mid queryTerms \mid \vee \mid resultSet \mid$

This method does not take into account the order of the results. Each result has the same weight, so if a retrieved document at the end of a result set is absent in the user profile, the system gives the same misuse warning as if it is at the beginning of a result set, i.e. with higher relevance ranking. Since our relevance feedback method does not need to take into consideration the order of the query terms, definition 1 works very well to define and measure the warning level in our misuse detection using relevance feedback method. However, for our first two methods, i.e. collection clustering and result set clustering, we need to define a different method that takes the ranking of the returned

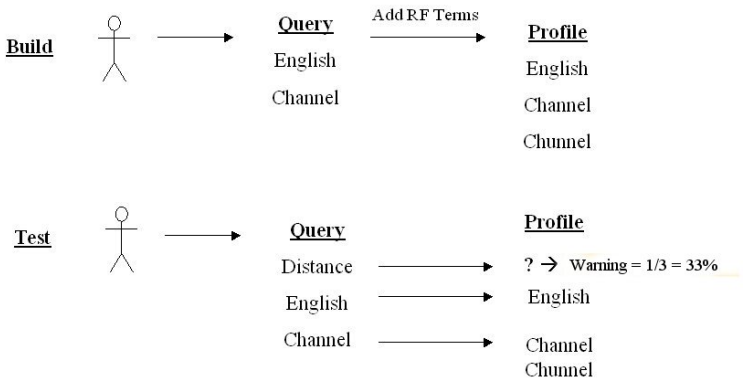

Figure 3: Relevance Feedback Method Process

documents into consideration. We now define a method that uses an index, i, to detect the order of missing documents from the user profile and take that into consideration in the final warning level.

Definition 2.

$$w = \frac{\left[\sum_{i \in I} \frac{s-i}{s}\right]}{A}$$

Where i= the index number of each result,
$I = \{i \mid x_i \in resultSet \cap x_i \notin profile\}$,
and $A =\mid I \mid$ and $s =\mid resultSet \mid$

This method does very well at calculating misuse warnings when the number of missing documents is small, however, when all of the documents are missing the misuse warning is not 1 as it should be, instead it is quite a bit less. Therefore, a more accurate way to measure the misuse warning might be to take the average of both methods in definitions 1 and 2 defined as definition 3.

Definition 3.

$$w = \frac{\frac{\left[\sum_{i \in I} \frac{s-i}{s}\right]}{A} + \frac{A}{s}}{2}$$

Where i=the index number of each result,
$I = \{i \mid x_i \in resultSet$ and $x_i \notin profile\}$
and $A =\mid I \mid$ and $s =\mid resultSet \mid$.

We define the misuse warning as in definition 3 for the first method, i.e. collection clustering, of misuse detection.

Furthermore, we define a misuse measurement for clustering query results. Each result set is measured against each cluster centroid to determine the closest cluster. The larger the similarity measure, the closer to the centroid. Thus the value of misuse warning measure must be closer to zero if the similarity is large, and closer to one, if the similarity measure is small.

Definition 4.

$$w = \left[1 - \frac{p+1}{\alpha \sum_{x \in R} max_{c \in C}[SC(x,c)]}\right]$$

Where $C = \{c \mid c \in$ clusters $\wedge c \in profile\}$,
$R = resultSet$,
$p = max(1, |\{x \mid x \in R$ AND $x \in C\}|)$,
$SC(x,c)$=the similarity measurement between x and c,
and α=a user defined weight used to put emphasis
on the Similarity Coefficient.

With definition 4 the larger the similarity measurement, the smaller the warning. The warning level is 0 if all the documents are included in the clusters.

Finally, for our final method of misuse detection we need to define a linear combination of warnings generated by the three methods. This measurement, i.e. fusion method, allows assigning weights to each of the three warnings to emphasize a particular method.

Definition 5.

$$w = \frac{\alpha \cdot a + \beta \cdot b + \gamma \cdot c}{\alpha + \beta + \gamma}$$

Where a=the misuse warning for method one
b = the misuse warning for method two,
c = the misuse warning for method three,
and α, β, and γ = arbitrary user-defined weights

4. EXPERIMENTAL FRAMEWORK AND RESULTS

We built and tested our misuse detection approach for information retrieval systems by using the four methods described in section 3, i.e. clustering, query results clustering, relevance feedback, and fusion methods.

We used a 2GB document collection from the Text Retrieval Conference(TREC) [11] as well as a smaller 10MB subset of the TREC Collection. The results of the small and big data collections were similar, thus we only present

Table 1: Misuse Warnings for Clustering Documents

Query files	1	2	3	4	5	6	7	8	9	10	11
Profiles											
1	0.0	0.78	0.66	0.63	0.77	0.71	0.51	0.75	0.47	0.65	0.78
2	0.80	0.0	0.73	0.63	0.82	0.85	0.73	0.75	0.81	0.89	0.84
3	0.87	0.87	0.0	0.83	0.75	0.72	0.90	0.83	0.76	0.86	0.80
4	0.85	0.87	0.72	0.0	0.81	0.90	0.71	0.82	0.81	0.88	0.84
5	0.82	0.89	0.84	0.85	0.0	0.84	0.79	0.84	0.86	0.87	0.89
6	0.72	0.83	0.84	0.90	0.37	0.0	0.59	0.84	0.47	0.74	0.76
7	0.65	0.80	0.78	0.82	0.77	0.71	0.0	0.57	0.47	0.65	0.76
8	0.65	0.78	0.87	0.64	0.78	0.70	0.71	0.0	0.51	0.65	0.78
9	0.81	0.95	0.89	0.91	0.81	0.73	0.78	0.84	0.0	0.85	0.83
10	0.82	0.87	0.89	0.82	0.82	0.81	0.87	0.84	0.67	0.0	0.84
11	0.83	0.92	0.89	0.84	0.84	0.84	0.84	0.85	0.73	0.65	0.0

Table 2: Misuse Warnings for Clustering Query Results

Query files	1	2	3	4	5	6	7	8	9	10	11
Profiles											
1	0.0	0.90	0.91	0.91	0.90	0.90	0.91	0.91	0.90	0.91	0.90
2	0.84	0.0	0.82	0.83	0.85	0.85	0.85	0.85	0.86	0.77	0.86
3	0.91	0.90	0.0	0.90	0.91	0.91	0.91	0.91	0.92	0.91	0.91
4	0.87	0.86	0.84	0.0	0.88	0.87	0.88	0.88	0.88	0.87	0.88
5	0.88	0.88	0.89	0.89	0.0	0.87	0.88	0.89	0.88	0.89	0.88
6	0.90	0.89	0.89	0.89	0.88	0.0	0.90	0.90	0.90	0.90	0.90
7	0.88	0.85	0.86	0.85	0.88	0.86	0.0	0.88	0.88	0.85	0.85
8	0.85	0.81	0.84	0.82	0.85	0.85	0.84	0.0	0.84	0.84	0.84
9	0.91	0.89	0.91	0.90	0.91	0.90	0.90	0.89	0.0	0.90	0.90
10	0.90	0.89	0.91	0.90	0.90	0.91	0.91	0.88	0.91	0.0	0.91
11	0.90	0.89	0.90	0.90	0.90	0.90	0.90	0.91	0.90	0.90	0.0

the results from the big collection. TREC does not contain any evaluation metrics for clustering or misuse detection. To evaluate the correctness of our results, we manually verified the result of all four proposed methods on a smaller data collection.

The experiments were performed on eleven profiles, each profile was tested against eleven query files consisting of eleven queries each. Thus the total number of tests run per method was $1,331$. The numbers in the tables represent the average misuse warning given when a specified query file is tested against a certain profile. The closer the number is to 1 the higher the warning level. We demonstrate the experimental results for all of our proposed methods in tables 1, 2, 3, and 4; and give the analysis on the results.

4.1 Experimental Results for Method 1: Clustering Documents

The results for clustering the documents are shown in Table 1. The misuse warnings are derived from definition 3. The warning is the ratio of the number of retrieved results absent in the profile clusters, considering the relevance ranking of the results. For example, the top 10 documents that are not in the profile generate a higher misuse warning compared to the to the next 10 documents. From these results, we can see that when the profile differs from the query file, the system returns warnings that are mainly in the 60% to 80% range. There are a few instances where the misuse warning for profiles tested against other files is in the mid $30-50\%$ range. The lower warnings are most likely caused by the overlap between the profile and the query file. When the query file matches the profile, however, the warning level is zero percent.

4.2 Experimental Results for Method 2: Clustering Query Results

The results for clustering the query results are shown in Table 2. The misuse warnings are derived from definition 4 with $\alpha = 2$. The larger the similarity coefficient, the closer the documents are to the centroid. By setting the $\alpha > 0$, we are putting emphasis on the closeness between documents. The results for this method are fairly consistent ranging in the upper 80% to the lower 90%. Overall, the misuse warnings for this type are much higher than our previous method.

4.3 Experimental Results for Method 3: Relevance Feedback

The results for our relevance feedback method are shown in Table 3. The misuse warnings are derived from definition 1. Overall the scores for this method are consistently in the $90-100\%$ range, except when the test query file matches the profile, in that case the warning value is predictably zero.

4.4 Experimental Results of Method 4: Fusion

The results for the fusion of all three methods is shown in Table 4. The misuse warnings are derived from definition 5 where α, β, and γ are all set equal to 1 to indicate that all three methods are used with an equal importance.

The results demonstrate that when the query test file is

Table 3: Misuse Warnings for Relevance Feedback

Query files	1	2	3	4	5	6	7	8	9	10	11
Profiles											
1	0.0	1.0	0.93	1.0	1.0	0.97	1.0	0.94	1.0	0.92	1.0
2	1.0	0.0	1.0	0.83	1.0	1.0	0.97	1.0	0.99	0.98	1.0
3	0.95	1.0	0.0	0.93	0.95	1.0	0.97	0.97	0.98	0.97	0.92
4	0.98	0.97	0.83	0.0	1.0	1.0	1.0	1.0	1.0	0.94	1.0
5	1.0	1.0	0.98	1.0	0.0	0.95	0.90	0.95	1.0	0.98	0.95
6	1.0	0.99	1.0	1.0	0.85	0.0	0.82	1.0	0.91	1.0	0.93
7	0.98	1.0	0.95	1.0	0.89	0.88	0.0	0.91	0.96	0.98	0.91
8	0.95	1.0	0.95	1.0	1.0	0.97	0.95	0.0	0.96	0.98	1.0
9	1.0	1.0	0.98	1.0	1.0	0.95	0.97	1.0	0.0	0.95	0.98
10	0.95	1.0	0.88	0.92	0.95	0.95	0.94	1.0	0.95	0.0	0.93
11	1.0	1.0	0.89	1.0	1.0	0.98	0.94	1.0	0.85	0.98	0.0

Table 4: Misuse Warnings for Fusion Method

Query files	1	2	3	4	5	6	7	8	9	10	11
Profiles											
1	0.0	0.89	0.83	0.85	0.89	0.86	0.81	0.87	0.79	0.83	0.89
2	0.89	0.0	0.85	0.76	0.89	0.90	0.85	0.87	0.89	0.88	0.90
3	0.91	0.92	0.0	0.89	0.87	0.88	0.93	0.90	0.89	0.91	0.88
4	0.90	0.90	0.80	0.0	0.90	0.92	0.86	0.90	0.90	0.90	0.91
5	0.90	0.92	0.90	0.91	0.0	0.89	0.86	0.89	0.91	0.91	0.91
6	0.87	0.90	0.91	0.93	0.70	0.0	0.77	0.91	0.76	0.88	0.86
7	0.84	0.88	0.86	0.89	0.85	0.82	0.0	0.79	0.77	0.83	0.84
8	0.82	0.86	0.89	0.82	0.88	0.84	0.83	0.0	0.77	0.82	0.87
9	0.91	0.95	0.93	0.94	0.91	0.86	0.88	0.91	0.0	0.90	0.90
10	0.89	0.92	0.89	0.88	0.89	0.89	0.91	0.91	0.84	0.0	0.89
11	0.91	0.94	0.89	0.91	0.91	0.91	0.89	0.92	0.83	0.84	0.0

different from the profile, the warnings from the three methods are smoothed out to be around 80% to 90%. When the query file matches the profile, however, the warning level is zero. It is important to note that by putting different weights on different methods, we can put more or less emphasis on the warning. In this way, we can increase the accuracy of the misuse warning.

4.5 Validation of Results

We measured the average accuracy of our misuse detection system by evaluating the top 10% warning generated for each test case. We assumed that these top warning are the ones that should trigger most of the attention to the user's activities. Our system achieved an average accuracy of 92% for detecting misuse. Figure 4 illustrates the comparison among the accuracy of the methods in our misuse detection system.

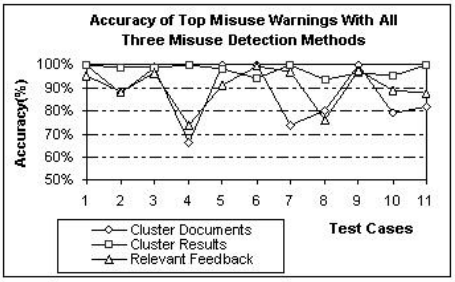

Figure 4: Clustering Query Results Process

5. CONCLUSION AND FUTURE WORK

We presented a novel approach to detect misuse within the information retrieval system by building user profiles based on user behavior. Any new activity of the user is compared to the user profile to detect a potential misuse for the authorized user. We proposed four different methods to detect misuse in information retrieval systems. Our results favorably demonstrated the validity of our approach.

Our current system does not take into account how many times a cluster, document, or term could be added to a profile. By taking this factor into consideration our original assumption–all queries a user submits during the building phase are correct–may be replaced with a new less general assumption–most of the queries a user submits during the building phase are correct. We also believe that setting different thresholds will effect the misuse warning. In the future, we plan to take into consideration these factors.

6. REFERENCES

[1] Steven M. Beitzel, Eric C. Jensen, Angelo J. Piloto, Nazli Goharian, and Ophir Frieder, *Parallelizing the buckshot algorithm for efficient dcoument clustering*, ACM 11th Conference on Information and Knowledge Management (CIKM) (2002).

[2] Christina Yip Chung, Michael Gertz, and Karl Levitt, *Demids: A misuse detection system for database systems*, In Third International IFIP TC-11 WG11.5 Working Conference on Integrity and Internal Control in Information Systems (1999), 159–178, Kluwer Academic Publishers.

[3] Ophir Frieder et al., *Detection of misuse of unauthorized access in an information retrieval system*, United States Patent Application Publication #20030037251 (2003).

[4] David A. Grossman and Ophir Frieder, *Information retrieval algorithms and heuristics*, 2 ed., Kluwer Academic Publishers, 2003.

[5] A.K. Jain, M.N. Murty, and P.J. Flynn, *Data clustering: a review*, ACM Computing Surveys **31** (1999), no. 3, 209–213.

[6] Sandeep Kumar and Eugene H. Spafford, *A pattern matching model for misuse intrusion detection*, Proceedings of the National Computer Security Conference (1994), 11–21.

[7] Terran D. Lane, *Machine learning techniques for the computer security domain of anomaly detection*, Ph.D. thesis, Department of Electrical and Computer Engineering, Purdue University, aug 2000.

[8] Yihua Liao and V. Rao Vemuri, *Using text categorization techniques for intrusion detection*, Tech. report, University of California, Davis, One Shields Avenue, Davis, CA 95616, 2002.

[9] C.X. Ling, J. Gao, H. Zhamg, W. Qian, and H. Zhang, *Improving encarta search engine performance by mining user logs*, International Journal of Pattern Recognition and Artificial Intelligence(IJPRAI) **16** (2002), no. 8.

[10] Jack Marin, Daniel Ragsdale, and John Surdu, *A hybrid approach to the profile creation and intrusion detection*, DARPA Information Survivability Conference and Exposition (DISCEX II'01) **1** (2001).

[11] National Institute of Standards and Technology, *Text retrieval conference(trec)*, December 2002, http://trec.nist.gov/.

[12] F. Salton, *Automatic text processing*, Addison Wesley, Massachusetts, 1989.

[13] Radu Sion, Mikhail Atallah, and Sunil Prabhakar, *On-the-fly intrusion detection for web portals*, Tech. Report 2002-36, Purdue University, West Lafayette, IN 47909, 2002.

[14] Alexandr Sleznyov and Oleksiy Mazhelis, *Learning temporal patterns for anomaly intrusion detection*, Symposium on Applied Computing (2000), 209–213.

[15] Ying Zhao and George Karypis, *Evaluation of hierarchical clustering algorithms for document datasets*, Proceedings of the eleventh international conference on Information and knowedge management (2002), 515–524.