Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5823439, 11 pages
https://doi.org/10.1155/2018/5823439

Research Article

A Data Leakage Prevention Method Based on the Reduction of Confidential and Context Terms for Smart Mobile Devices

Xiang Yu,[1] Zhihong Tian,[2] Jing Qiu,[2] and Feng Jiang[3]

[1]*School of Electronics and Information Engineering, Taizhou University, Taizhou 318000, China*
[2]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[3]*College of Science and Technology, Harbin Institute of Technology, Harbin 150001, China*

Correspondence should be addressed to Zhihong Tian; tianzhihong@gzhu.edu.cn and Jing Qiu; qiujing@gzhu.edu.cn

Received 27 April 2018; Revised 30 August 2018; Accepted 24 September 2018; Published 21 October 2018

Guest Editor: Ding Wang

Early data leakage protection methods for smart mobile devices usually focus on confidential terms and their context, which truly prevent some kinds of data leakage events. However, with the high dimensionality and redundancy of text data, it is difficult to detect the documents which contain confidential contents accurately. Our approach updates cluster graph structure based on CBDLP (Data Leakage Protection Based on Context) model by computing the importance of confidential terms and the terms within the range of their context. By applying CBDLP with pruning procedure which has been validated, we further remove the redundancy terms and noise terms. Actually, not only can confidential terms be accurately detected but also the sophisticated rephrased confidential contents are detected during the experiments.

1. Introduction

With the development of Internet and information technology, smart mobile devices appear in our daily lives, and the problem of information leakage on smart mobile devices will follow which has become more and more serious [1, 2]. All kinds of private or sensitive information, such as intellectual property and financial data, might be distributed to unauthorized entity intentionally or accidentally. And that it is impossible to prevent from spreading once the confidential information has leaked.

According to survey reports [3, 4], most of the threats to information security are caused by internal data leakage. These internal threats consist of approximate 29% private or sensitive accidental data leakage, approximate 16% theft of intellectual property, and approximate 15% other thefts including customer information, and financial data. Further, the consensus of approximate 67% organizations shows that the damage caused from internal threats is more serious than those form outside.

Although laws and regulations have been passed to punish various behaviors of intentional data leakage, it is still hard to prevent data leakage effectively. Confidential data can be easily disguised by rephrasing confidential contents or embedding confidential contents in nonconfidential contents [5, 6]. In order to avoid the problems arising from data leakage, lots of software and hardware solutions have been developed which are discussed in the following chapter.

In this paper, we present CBDLP, a data leakage prevention model based on confidential terms and their context terms, which can detect the rephrased confidential contents effectively. In CBDLP, a graph structure with confidential terms and their context involved is adopted to represent documents of the same class, and then the confidentiality score of the document to be detected is calculated to justify whether confidential contents is involved or not. Based on the attribute reduction method from rough set theory, we further propose a pruning method. According to the importance of the confidential terms and their context, the graph structure of each cluster is updated after pruning. The motivation of the paper is to develop a solution which can prevent intentional or accidental data leakage from insider effectively. As mixed-confidential documents are very common, it is very important to accurately detect the documents containing confidential contents even when most of the confidential contents have been rephrased.

The remainder of this paper is organized as follows. In Section 2, we introduce previous related work on data leakage prevention. In Section 3, we present CBDLP model together with the corresponding clustering, decision, and calculation algorithms. The experiments conducted to evaluate CBDLP in all circumstance are discussed in Section 4. Finally, Section 5 concludes this paper and discusses the directions of our future research.

2. Related Work

In this section, we review clustering of textual documents, attribute reduction method, and graph representation of textual documents, respectively.

2.1. Clustering of Textual Documents. The problem of clustering textual documents is similar to high dimensional clustering. In general, each term of a textual document is considered as an independent dimension and then each document is considered as a vector consists of thousands of terms. By calculating the angle cosine measure between documents, textual documents can be classified in terms of similarity which is reflected by the angle cosine value [7–10].

Vector space model, VSM, is one of the most widely used text representation models [11], which is first presented by Salton in the 1960s and successfully applied in SMART, a system for the manipulation and retrieval of text. In VSM model, a textual document is represented as $D = D((T_1, W_1), (T_2, W_2) \ldots, (T_n, W_n))$, where T_i and W_i denote the ith term and its weight in the document, respectively, and then the classification of a textual document is determined by calculating the similarity between the textual document to be classified and the textual documents whose classification are already known.

Term frequency and inverse document frequency, TF-IDF, is a frequently employed and effective statistics method which is used to evaluate the importance of a term for a documents collection [12]. As is well known, the importance of a term is proportional to the frequency of its occurrence in a document and is inversely proportional to the frequency of its occurrence in the whole corpus. Till now, TF-IDF has been widely used in various fields, such as text mining, search engine, and information retrieval.

On the basis of VSM model and TF-IDF method, existing textual documents clustering algorithms can be divided into five main categories: partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. Partitioning methods, which are efficient and insensitive to the sequence of documents, divide n documents into k clusters in terms of clustering criteria. The representative partitioning methods include k-means and k-medoids [13, 14]. Hierarchical methods disintegrate documents into different clusters or integrate different documents together into one cluster in terms of the similarity with a top-down or bottom-up hierarchical manner. The representative hierarchical methods include BIRCH and CURE [15, 16]. Other than partitioning methods, density-based methods focus on the density of a certain area. When the density of the documents within a certain area exceeds a predefined threshold, they are incorporated into the same cluster. The representative density-based methods include DBSCAN and OPTICS [17, 18]. Grid-based methods partition data space into limited cells in advance and integrate adjacent cells whose density exceed the density threshold into the same cluster. The representative grid-based methods include STING and CLIQUE [19, 20]. In model-based methods, different models are bound up with each cluster respectively, and the objective is to find all data subsets that fit each model best. Statistics solution, such as SVM [21], and neural network solution are adopted extensively in model-based methods [22]. The support vector clustering algorithm created by Hava Siegelmann and Vladimir Vapnik applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

In this paper, we calculate the angle cosine value which reflects the similarity between documents and cluster documents with DBSCAN. DBSCAN, proposed by Martin Ester in 1996, is a density-based clustering algorithm which is widely cited in scientific literature [23], and it is awarded the test of time award in 2014 [24]. When clustering, other than k-means, DBSCAN does not need to specify the number of clusters and it can find the clusters of arbitrary shape. In addition, DBSCAN is robust to outliers as opposed to k-means.

2.2. Graph Representation of Textual Documents. Graphs have already been used in many text-related tasks, which employ graph as the model for text representation instead of the existing methods [25]. As an alternative method to the vector space model for representing textual information, graphs can be created from documents and be further used in the text-related tasks such as information retrieval [26], text mining [27], and topic detection [28].

In general, graph-based model is usually employed in the domain of information retrieval such as PageRank [29] and HITS [30]. When determining the similarity, graph matching, which is generally used to detect similar documents, is NP in complexity [31], whereas the methods based on vector space model perform efficiently by calculating the Euclidean distance or Cosine measure between document vectors [32]. The main advantage of graph-based model is that it can not only capture the contents and structure of a document but also represent the terms together with their context. To the best of our knowledge, graph-based model is seldom employed in text-related tasks. Schenker presents a graph-related algorithm with its several variants [33] in which a graph is presented and the terms connected with edges are considered as nodes. The differences between the variants are related to term-based techniques. Gilad Katz presents CoBAn, a context based model for data leakage prevention, which enlightens us a lot [34]. However, CoBAn is partly influenced by the limitation of k-means which is employed in CoBAn. Moreover, there might exist some redundancy nodes in the graph generated in CoBAn. Xiaohong Huang et al. propose an Adaptive weighted Graph Walk model (AGW) to

solve the problem of transformed data leakage by mapping it to the dimension of weighted graphs [35].

In this paper, we employ a hybrid approaches which combines graph and vector representations. When clustering documents, we employ DBSCAN with cosine measure. When representing the confidential textual content and its context of each cluster, the graph of each cluster which includes only confidential and contextual nodes is created.

2.3. Redundancy Information Reduction. When dealing with text-related tasks, redundancy information is generally useless and even worse, it might decrease the efficiency of task execution. There exist many representative redundancy information reduction methods such as PCA [36], SVD [37], LSI [38], etc. The principle of PCA is to transform multiple attributes into a few primary attributes, which can reflect the information of original data effectively. However, the complexity of PCA is generally high and there might be part of original information loss. More than characteristics, SVD has almost the same advantages and disadvantages as PCA does. LSI represents textual data with latent topics that consists of specific terms, but in most cases, the influence of specific terms are ignored. In this paper, the reduction method from rough set theory, as shown in Section 3, is employed and partly recomposed to meet requirements.

2.4. Data Leakage Prevention. With the number of leakage incidents and the cost they inflict continues to increase, the threat of data leakage posed to companies and organizations has become more and more serious [39–41]. Considering the enormity of data leakage prevention, various models and approaches have been developed to address the problem of data leakage prevention. Tripwire is a more recent prototype system proposed by Joe DeBlasio et al. in 2017; it registers honey accounts with individual third-party websites, and thus access to an email account provides indirect evidence of credentials theft at the corresponding website [42]. However, Tripwire is more suitable for forensics rather than confidential data leakage prevention. In 2018, Wenjia Xu et al. propose a new promising image retrieval method in ciphertext domain by block image encrypting based on Paillier homomorphic cryptosystem which can manage and retrieve ciphertext data effectively [43]. Nevertheless, the method focus on data encryption rather than data detection. Since smart devices based on ARM processor become an attractive target of cyberattacks, Jinhua Cui et al. present a scheme named SecDisplay for trusted display service in 2018, it protects sensitive data displayed from being stolen or tampered surreptitiously by a compromised OS [44]. But it pays less attention to the scenarios of intentional or accidental data leakage from insider. According to the work of Ding Wang et al., lots of authentication schemes have been proposed to secure data access in industrial wireless sensor networks, however, they do not work well [45]. In addition, Ding Wang et al. develop a series of practical experiments to evaluate the security of four main suggested honeyword-generation methods and further prove that they all fail to provide the expected security [46].

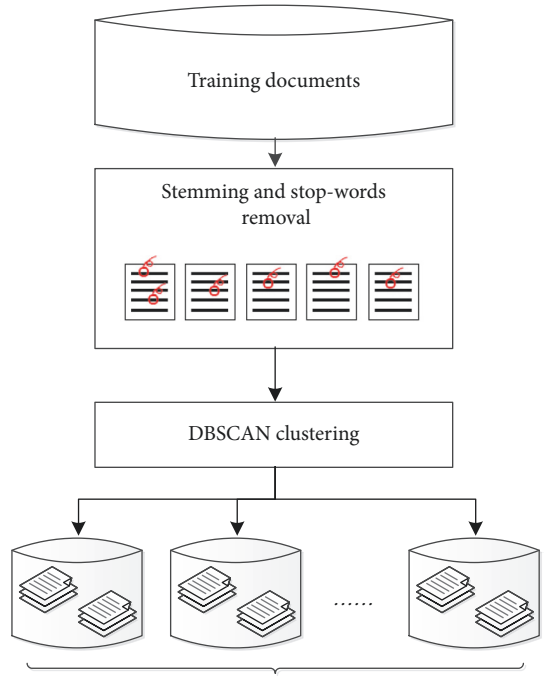

Figure 1: DBSCAN method.

3. CBDLP Model

CBDLP consists of training phrase and detection phrase. The training phrase can be further divided into three steps, clustering step, graph building step, and pruning step. During the training phrase, the training documents are first classified into different clusters, then each cluster is represented by graph, and finally the nodes of each graph are pruned in terms of their importance. During the detection phrase, documents are matched to the graphs of clusters respectively and the confidential scores are calculated. A document is considered as confidential only if its confidential score exceeds a predefined threshold. The detail of CBDLP training phrase is presented in Algorithm 1.

3.1. Clustering Documents with DBSCAN. In the first step, we apply stemming and stop-words removal to all documents in training set, and transform the processed documents into vectors of weighted terms. After applying DBSCAN with cosine measure to the vectors, which represent the training documents, each resulting cluster represents an independent topic of training documents and there might exist both confidential and nonconfidential documents. As shown in Figure 1.

The procedure of DBSCAN is described as follows:

Step 1. A data set is given with n documents and ε as the threshold of minimal similarity between documents of the same cluster and $MinPts$ as the threshold of minimal number of documents in a cluster.

Input: C - Confidential documents set
N - Non-confidential documents set
TR_{min} - The minimum similarity threshold
Output: CR - The set of clusters, each with the centroid and corresponding graph
CT - The set of confidential terms in clusters
$ContextT$ - The set of context terms

(1) $T \longleftarrow C \cup N$
(2) $CR \longleftarrow DBSCAN\,(T)$ % The result of clustering T is saved in CR
(3) Initializing $CT[CR]$ %The scores of confidential terms are saved in CT
(4) Initializing $ContextT$ %The context terms set of each confidential term is saved in $ContextT$
(5) **for** (each cr in CR)
(6) { Calculate the similarity between cr and the other clusters
(7) Create language model for cr, and calculate the scores for each confidential term
(8) $TR \longleftarrow$ initial the threshold of cluster similarity
(9) **while** $(TR > TR_{min})$
(10) { $C_{temp} \longleftarrow$ All clusters whose similarity to $cr > TR$
(11) Create language model for the documents of C_{temp}
(12) $CT[cr] \longleftarrow$ Based on new language model, Update the scores of confidential terms
(13) **for**(each confidential term ct in cr)
(14) { Detect the occurrence of ct in $C \cup N$
(15) $P_{confidential\_doc}(term, ct) \longleftarrow$ For each context term of ct, calculate the probability of the appearance both ct and the context term in confidential documents.
(16) $P_{non\_confidential\_doc}(term, ct) \longleftarrow$ For each context term of ct, calculate the probability of the appearance both ct and the context term in non-confidential documents.
(17) $ContextT \longleftarrow$ Calculate the value of $P_{confidential\_doc}(term, ct)/P_{non\_confidential\_doc}(term, ct)$ for each confidential term ct
(18) Detect all clusters whose similarity is greater than TR, and detect the occurrences of all terms in the clusters.
(19) $ContextT \longleftarrow$ Update the probability of the context terms that appear in the scopes of different confidential terms
(20) }
(21) Reduce the value of TR
(22) }
(23) }

Algorithm 1: CBDLP.

Step 2. Start with an arbitrary document that has not been visited and find all the documents in its ε-neighborhood. If the number of documents in the neighborhood exceeds $MinPts$, incorporate the documents into the same cluster and label them.

Step 3. If not all documents have been visited, start from another arbitrary document which has not been visited.

Step 4. Mark the documents which are not labelled as noise.

3.2. Representing Clusters with Graph. In this step, the confidential contents in all clusters, which include not only the confidential terms but also their context, need to be represented by graphs. The procedure of creating graph representation for the clusters which include confidential contents is described as follows:

(1) Detect the confidential terms provided by domain experts or inferred from the key terms of training documents.

(2) Analyze the context of each confidential terms.

(3) Create the graph representation for confidential terms and their contexts on the cluster level.

3.2.1. Detect Confidential Terms. In general, a term, which appears in confidential documents with high probability and appears in nonconfidential documents with low probability, is considered as confidential term. We first build language models for the confidential and non-confidential documents of the same cluster, which are denoted by cVM (confidential vector model) and $ncVM$ (nonconfidential vector model). Then the confidentiality score can be represented by the ratio of its probability in confidential documents to that in non-confidential documents as shown in as follows, where $P_{cVM}(t)$ and $P_{ncVM}(t)$ denote the probability of term t in confidential and nonconfidential language models, respectively:

$$\forall t \in cVM,$$
$$score+ = \frac{P_{cVM}(t)}{P_{ncVM}(t)} \tag{1}$$

However, there may exist the following problem. If a cluster includes only few nonconfidential documents or possibly none at all, its language model cannot fully represent the nonconfidential documents in it. The solution we proposed follows an expanding manner; we first predefine the minimal similarity threshold TR_{min} and iteratively expand the $ncVM$ to include more clusters. TR is referred to as the similarity threshold of cosine measure of the cluster with few

nonconfidential documents. Note that not all clusters need to be expanded. After each iteration, we lower the value of TR. Unless TR is greater than TR_{min}, the nonconfidential documents of the expanding clusters are included to recalculate the scores of terms in original cluster.

When the adjacent clusters are included and the scores of confidential terms are recalculated, each term whose score is greater than 1 is considered as confidential term, which means the term is more likely to appear in confidential documents than in non-confidential documents. After this phase, the set of confidential terms, CT, is obtained.

3.2.2. Analyze the Context of Confidential Terms. After confidential terms detection, we further analyze the context of confidential terms. Apparently, a term is more likely to be considered as confidential if it appears in the similar contexts in other confidential documents. Inversely, if the context of a confidential term frequently appears in nonconfidential documents, the probability of the confidential term being part of confidential contents is lower.

As a predefined parameter, context span η determines the number of terms that precede and follow the confidential term. Context span with high value might increase the computational cost, inversely, and context span with low value could not provide adequate context information of confidential terms. Experimental results show that $\eta = 10$ tends to be the optimal value of context span in our experiments, which means that the context of a confidential term consists of the five terms preceding it and the other five following it. Apparently, only the context of the confidential terms in confidential documents needs to be taken into account.

The probabilities of a confidential term together with its context appearing in confidential documents and nonconfidential documents, which are denoted by $P_c(key_{context}/key)$ and $P_{nc}(key_{context}/key)$, are calculated separately. If the former is higher than the latter, the corresponding confidential contents can be well represented by the confidential term with its context. $P_c(key_{context}/key)$ is defined as the number of confidential documents in which the confidential term with its context appears divided by the number of confidential documents in which only the confidential term appears. And $P_{nc}(key_{context}/key)$ is defined as the number of nonconfidential documents in which the confidential term with its context appears divided by the number of nonconfidential documents in which only the confidential term appears.

As mentioned above, we predefine the similarity threshold of minimum cosine measure TR_{min}, and iteratively expand to include more clusters. TR is referred to as the similarity threshold of cosine measure between the cluster with few non-confidential documents and its expanding cluster. After each iteration, we lower the value of TR at a certain rate μ which is predefined, namely $TR = \mu^* TR$. Unless TR is greater than TR_{min}, the non-confidential documents of the expanding cluster are included to recalculate the scores of context terms in original cluster. By including more adjacent clusters, we can accurately estimate which terms are most likely to indicate the confidentiality of the document.

By subtracting the probability of the appearance of each context term with confidential term in non-confidential documents from the probability of the appearance of them in confidential ones, the score of each context term is calculated, as shown in (2).

The reason for employing subtraction rather than division is to avoid large fluctuations in the values of the context terms. When employing division, even a single document can dramatically change the probabilities as only the documents including confidential terms are taken into account.

$$score+ = P_c\left(\frac{key_{context}}{key}\right) - P_{nc}\left(\frac{key_{context}}{key}\right) \quad (2)$$

We iteratively expand to include more clusters. After each iteration, we lower the value of TR until TR is less than TR_{min}, and the score of each context term is calculated, as shown in (3) in which n_{clu} denotes the number of clusters involved.

$$score+ = \frac{1}{n_{clu}}\left(P_c\left(\frac{key_{context}}{key}\right) - P_{nc}\left(\frac{key_{context}}{key}\right)\right) \quad (3)$$

After this phase, the set of context terms with their scores, $ContextT$, is obtained. For each confidential term, its context terms whose scores are positive are more likely to appear in confidential documents with the confidential term.

3.2.3. Create Graph Representation. After the operations described in previous section, confidential terms and their context can be easily represented as nodes and connected together according to their interrelation. As shown in Figure 2, for each cluster, a set of confidential terms and a set of its context terms are obtained after the training phase, and confidential terms and its context terms are represented as confidential nodes and context nodes respectively. Confidential nodes are connected together as long as there exists at least one common context node between them.

3.3. Pruning Nodes of Graph. Due to the calculation of confidential terms and their context terms are based on statistics scores, there might exist occasional case of a nonconfidential term with high score because of term abuse. In the pruning phase, we employ the method of term reduction in rough set theory to remove the redundancy nodes in graph.

With the information of confidential and nonconfidential documents, we evaluate the importance of nodes in graph for each cluster. A node in graph can be pruned only if the removal of the term represented by the node does not influence the results of identifying the confidential documents in this cluster. As shown in (4), $Impt(t_i)$ denotes the importance measure of term t_i which is represented as node i in graph. And $Pos_G(C)$ denotes the portion in confidential documents set C can be identified correctly by graph G. Similarly, $Pos_{\{G-n_i\}}(C)$ denotes the portion in confidential documents set C can be identified correctly by graph $\{G-n_i\}$, which means node n_i is removed from graph G. The detail of the pruning procedure for graph is presented in Algorithm 2.

$$Impt(t_i) = \frac{\left(\left|Pos_G(C)\right| - \left|Pos_{\{G-n_i\}}(C)\right|\right)}{|C|} \quad (4)$$

Input: C - Confidential documents set
N - Non-confidential documents set
CR - Clusters set, each with the centroid and corresponding graph
$ContextT$ - The set of context terms
Output: CT' - The updated set of confidential terms in clusters
$ContextT'$ - The updated set of context terms
(1) $irr\_terms \longleftarrow CT \cup ContextT$
(2) **for** (each t in irr_terms)
(3) **If** $(Impt(t) = (|Pos_T(C_{cr} \wedge N_{cr})| - |Pos_{T-t}(C_{cr} \wedge N_{cr})|)/|cr| < Impt\_threshold)$
(4) $irr\_terms = irr\_terms - t$
(5) **for** (each t in irr_terms)
(6) { $CT' \longleftarrow$ Update confidential terms in CT
(7) $ContextT' \longleftarrow$ Update context terms in $ContextT$ }

Algorithm 2: Pruning.

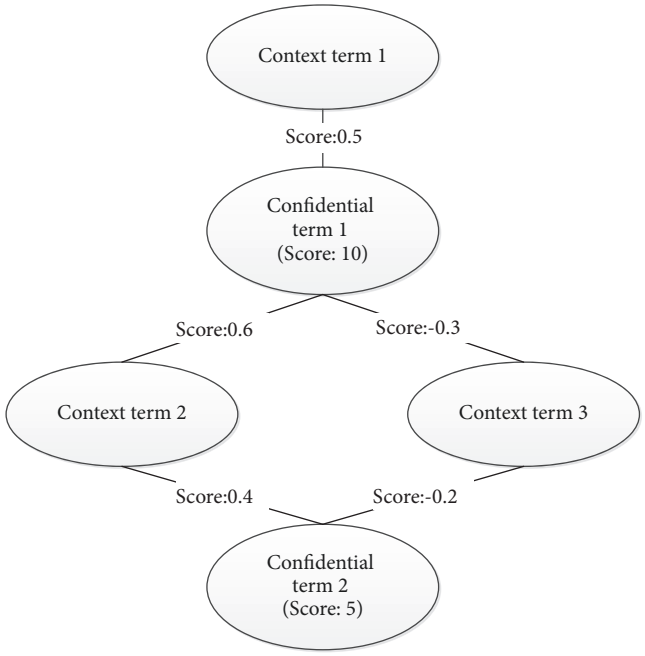

Figure 2: An example of confidential nodes connected through their context nodes.

3.4. Detection Phrase. Obviously, a confidential document without any modification is easy to be detected according to confidential terms. However, the confidential documents, which are rephrased or partitioned into portions and further concealed in different nonconfidential documents as most plagiarizers often do, can hardly be detected. Once the confidential contents detection fails, it is more likely to lead to data leakage or copyright infringement.

In the detection phase, we employ CBDLP model to deal with three scenarios that could possibly happen. The three different scenarios are described as follows:

(i) Each confidential document is detected as a whole.

(ii) Each confidential document is divided into portions and embedded in nonconfidential ones.

(iii) The confidential terms in confidential documents are rephrased completely.

The detection method we employed includes three steps as shown in Figure 3, which are described as follows:

(1) Classify the documents to be tested to the corresponding clusters.

(2) Identify the confidential terms and their context terms according to the graphs of the corresponding clusters.

(3) Calculate the confidentiality scores for the documents and draw the conclusion that whether a document is confidential or not.

Then, the security model, which combines the training phrase and the detection phrase, is shown in Figure 4.

4. Experiments

In this section, we evaluate the performance of CBDLP on Reuters-21578 dataset. As testing dataset, Reuters-21578 consists of 21578 pieces of news distributed by Reuters in 1987 which are saved in 22 files. Reuters-21578 dataset is manually classified as five categories, each of which can be subdivided into different number of subcategories. For example, the news of economy includes the inventories subset, gold subset, and money-supply subset.

4.1. Performance Experiments. In the experiments, we present the data leakage prevention method based on CBDLP model, and also present a modified model without pruning step which is represented as CBDLP-Pr. Since SVM has been proved to be an excellent classifier with high accuracy and CoBAn performs well in the scenario where confidential contents are embedded in nonconfidential documents or rephrased, we compare the performance of CBDLP, CBDLP-Pr, SVM, and CoBAn. We evaluate the performance of the methods in this paper with true positive rate (TPR) and false positive rate (FPR), and our goal is to maximize TPR and minimize FPR concurrently.

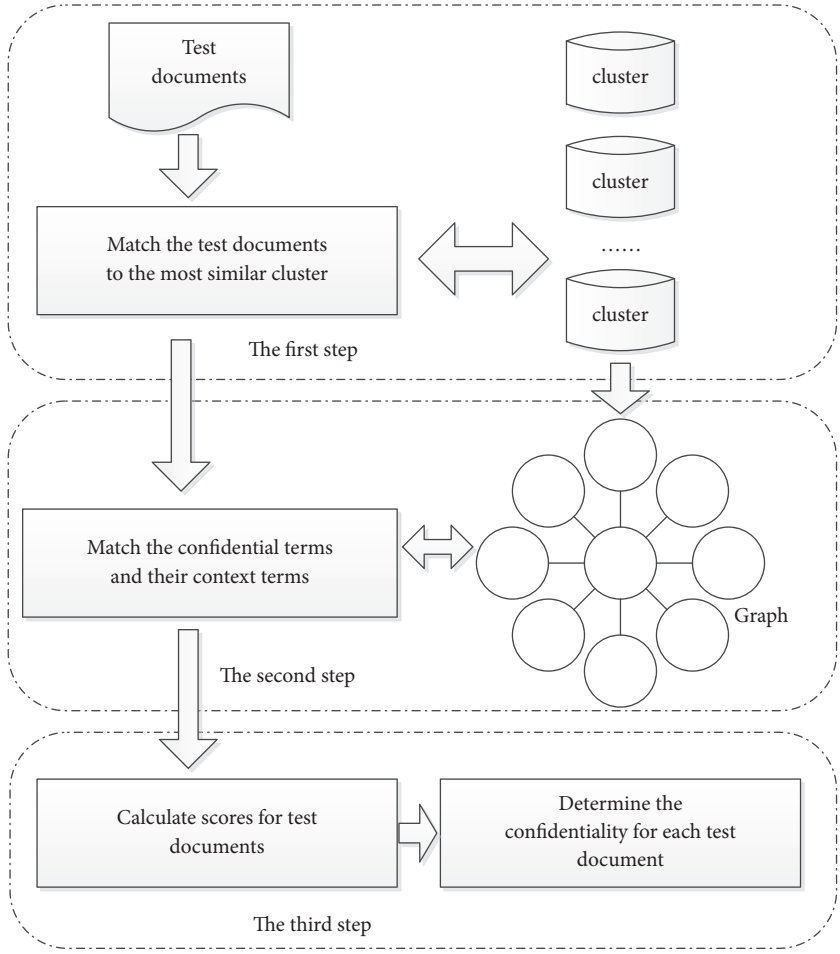

Figure 3: The detection of test documents.

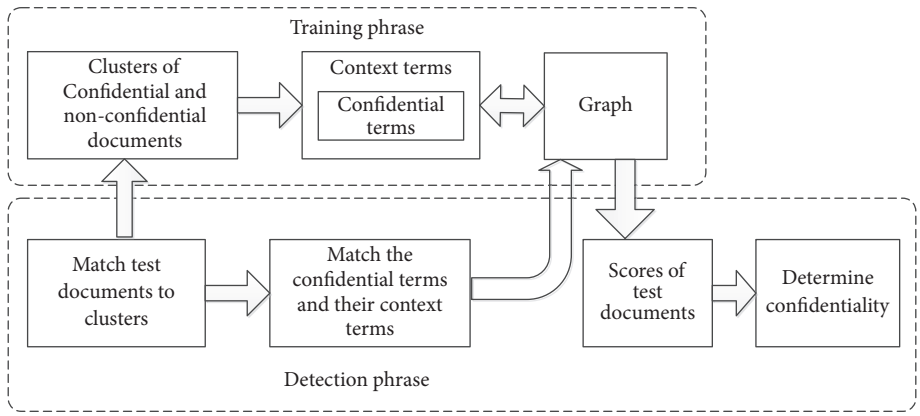

Figure 4: The security model.

We conduct experiments on the three scenarios which are described above. As for the first type of scenario, we select the news of "earn" as the carrier for confidential contents and mix them with the news from other economy subsets as training dataset and testing dataset separately. As for the second type of scenario, we extract the contents from the documents of "earn" subset and embed them in the documents from other subsets. The embedded portions are detected as confidential contents. As for the third type of scenario, we manually rephrase the contents in the documents of "earn" subset and embed them in the documents from other subsets.

4.1.1. Confidential Documents as a Whole. The experimental result of the first scenario is presented in Figure 5. As shown in

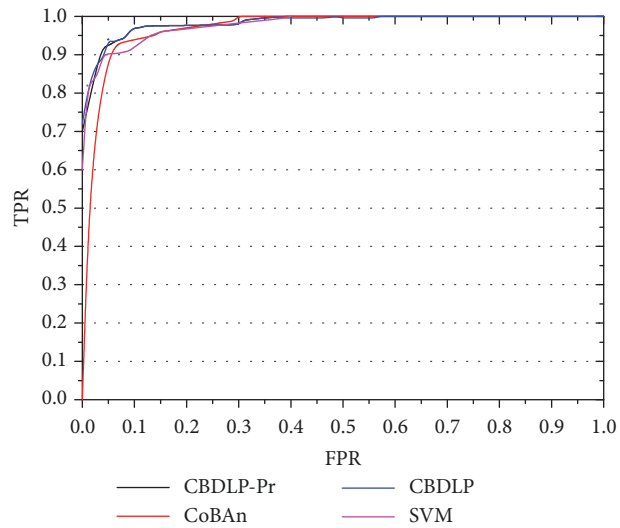

Figure 5: Performance of detecting confidential documents as a whole.

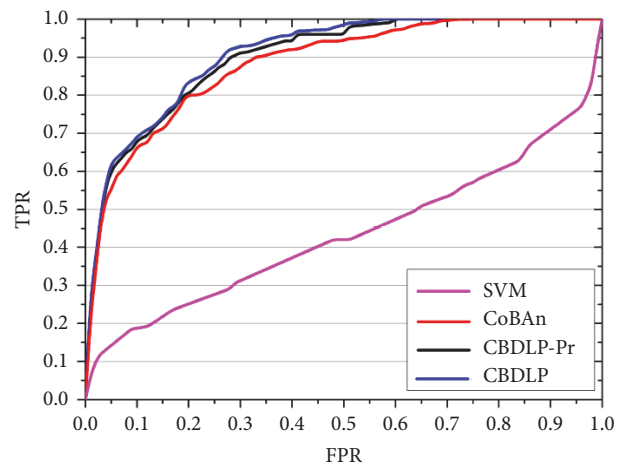

Figure 6: Performance of detecting the confidential contents embedded in nonconfidential documents.

Figure 5, when dealing with the scenario where confidential documents are considered as a whole, the performance of the four detection algorithms has no much difference. In spite of that, CBDLP and CBDLP-Pr still perform slightly better than CoBAn and SVM, which can be explained as that the performance of CoBAn is partly influenced by the limitation of k-means that it cannot deal with the clusters of various shapes effectively, and SVM only focuses on the confidential terms nevertheless ignores the context terms. In this scenario, since the documents containing confidential terms are explicitly detected as confidential documents, the performance of the four methods has no much difference.

4.1.2. Confidential Portions Embedded in Nonconfidential Documents. The result of the second scenario is presented in Figure 6. As shown in Figure 6, when dealing with the scenario where the confidential portions are embedded in nonconfidential documents, CBDLP, CBDLP-Pr, and CoBAn

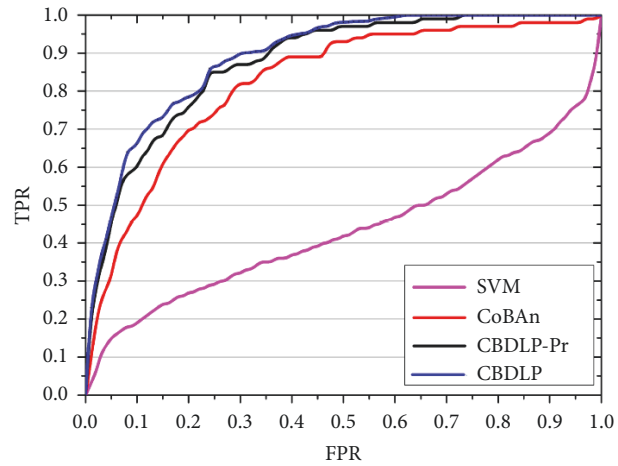

Figure 7: Performance of detecting the rephrased confidential contents embedded in nonconfidential documents.

perform better than SVM, which can be explained as that SVM is deceived by the scenario due to its statistics nature. As expected, the performance of CBDLP is slightly better than CBDLP-Pr and CoBAn due to its pruning step which removes the redundancy nodes in graph that might deteriorate the results of detection.

In this scenario, confidential portions are extracted from the documents defined as confidential and then embedded in the nonconfidential documents whose length are at least ten times larger than the extracted portions. Due to the statistical nature, most documents containing confidential portions are incorrectly detected as nonconfidential by SVM, which result in dramatic decline in the accuracy of SVM. Other than SVM, CBDLP, CBDLP-Pr, and CoBAn take the confidential terms together with their context into account, and most nonconfidential documents containing embedded confidential portions are detected as confidential.

4.1.3. Rephrased Confidential Contents in Nonconfidential Documents. The result of the third scenario is presented in Figure 7. As shown in Figure 7, when dealing with the scenario where the confidential contents are rephrased and embedded in nonconfidential documents, the performance of SVM deteriorates considerably due to its statistics nature. Since the rephrased contents do not deviate much from its original meaning, CBDLP, CBDLP-Pr, and CoBAn perform well. In addition, the performance of CBDLP is better than CBDLP-Pr and CoBAn due to its pruning step which removes the redundancy nodes in graph.

In this scenario, the rephrased confidential terms are embedded in nonconfidential documents which confuse SVM greatly, and most documents containing rephrased confidential contents are incorrectly detected as nonconfidential. Other than SVM, with the context of confidential terms taken into account, CoBAn detects most documents containing confidential contents; however, the accuracy of CoBAn is partially influenced by the cluster's terms graph which depends on the quality of clusters generated by k-means. As a result, CBDLP clusters documents with DBSCAN which

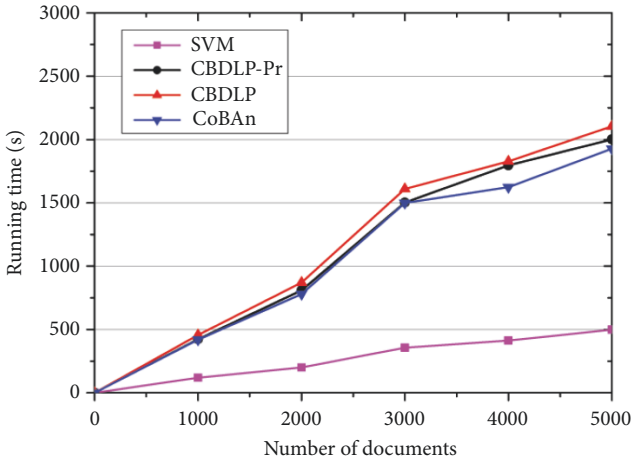

Figure 8: The scores of confidential documents and nonconfidential documents.

improves the quality of clusters and the cluster's terms graph; meanwhile, the pruning method removes the redundancy nodes in graph and further improves the performance of CBDLP.

4.2. Running Time Comparisons. In this experiment, we mixed non-confidential documents together with the three type of confidential documents, which are the whole confidential documents, the confidential contents embedded in nonconfidential documents and the rephrased confidential contents embedded in non-confidential documents. The experiment is conducted by using 10 fold cross validation. To Compare the running time of CBDLP, CBDLP-Pr, CoBAn, and SVM, we conduct the experiment on the datasets of different size. The result is as shown in Figure 8, the running time of training phase and testing phase are exhibited as line graph in which the running time of CBDLP, CBDLP-Pr, CoBan and SVM increase as more documents are added to the dataset. Although the additional steps of CBDLP, CBDLP-Pr, and CoBan result in more running time than SVM needs, their running time is still an order of magnitude; more than that, CBDLP performs much better than SVM does.

5. Conclusion and Future Work

In this paper, we present a new method for data leakage Prevention based on CBDLP model, which has the following advantages:

(1) It clusters the documents with DBSCAN and cosine measure which have been verified to be effective.

(2) It represents confidential terms and their context terms in graph.

(3) It presents a pruning method based on the attribute reduction method of rough set theory.

Up to now, some designated commercial DLP solutions can reduce the risk of most accidental leakage; however, they cannot provide sufficient protection against intentional leakage. And the other DLP solutions, such as firewalls, IDS, antimalware software, and management policies, which can provide assistance in detection intrusion or malicious software and enforce policies to protect data, still do not prevent intentional leaks perfectly. To the best of our knowledge, there might be two main future research topics on DLP, data leakage from mobile devices and accidental data leakage by insider.

Since accidental data leakage may be part of a larger attack in which their role will be mainly to activate an advanced persistent threat inside the organization, it is expected to continue to be one of the most challenging research topics. And our future work will focus on accidental data leakage in two directions. First, try to improve the efficiency and effectiveness of CBDLP on confidential contents detection. Second, adjust the model dynamically according to the changes of training dataset.

Data Availability

All data generated or analysed during this study are included in this published article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research is supported by the National Natural Science Foundation of China under Grant nos. 61871140 and 61572153.

References

[1] A. Goyal, F. Bonchi, and V. S. Lakshmanan, "On minimizing budget and time in influence propagation over social networks," *Social Network Analysis and Mining*, pp. 1–14, 2012.

[2] C. Aggarwal, *Social Network Data Analytics*, Springer, Berlin, Germany, 2011.

[3] "Information week global security survey," Information Week, 2004.

[4] D. Alassi and R. Alhajj, "Effectiveness of template detection on noise reduction and websites summarization," *Information Sciences*, vol. 219, pp. 41–72, 2013.

[5] D. Holmes, *Using language models for information retrieval [Ph.D. Thesis]*, Center for telematies and information technology, University of Twente, 2001.

[6] R. Böhme, "Security metrics and security investment models," in *Proceedings of the 5th international conference on advances in information and computer security*, 2010.

[7] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, Boston, Mass, USA, 1990.

[8] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Signal Processing*, vol. 35, no. 3, pp. 400-401, 1987.

[9] R. Jin, L. Si, A. G. Hauptmann, and J. Callan, "Language model for IR using collection information," in *Proceedings of the the 25th annual international ACM SIGIR conference*, pp. 419-420, 2002.

[10] W. W. Cohen, "Learning rules that classify e-mail," in *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pp. 18–25, 1996.

[11] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the Association for Information Science and Technology*, vol. 41, no. 6, pp. 391–407, 1990.

[13] C. Ordonez, "Clustering binary data streams with K-means," in *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, pp. 12–19, USA, June 2003.

[14] B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan, "Maintaining variance and k-medians over data stream windows," in *Proceedings of the Twenty second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2003*, pp. 234–243, June 2003.

[15] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: a new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.

[16] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," in *Proceedings of 1998 ACM SIGMOD International Conference Management of Data*, pp. 73–84, 1998.

[17] J. W. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.

[18] A. Hinneburg and D. A. Keim, "Optimal grid-clustering: towards breaking the curse of dimensionality in high-dimensional clustering," in *Proceedings of the 25th VLDB Conference*, pp. 506–517, 1999.

[19] W. Wang, J. Yang, and R. Muntz, "Sting: a statistical information grid approach to spatial data mining," in *Proceedings of the 23rd VLDB Conference*, pp. 186–195, 1997.

[20] R. Agrawal, J. Gehrke, and D. Gunopulos, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 94–105, 1998.

[21] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the KDD 2000*, pp. 169–178, ACM, New York, NY, USA, 2000.

[22] R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," *Information Sciences*, vol. 382-383, pp. 96–114, 2017.

[23] F. Jiang, Y. Fu, B. B. Gupta et al., "Deep learning based multichannel intelligent attack detection for data security," *IEEE Transactions on Sustainable Computing*, 2018.

[24] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the the 2000 ACM SIGMOD international conference*, pp. 439–450, May 2000.

[25] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[26] Salton, "Automatic text processing: the transformation, analysis and retrieval of information by computer," Tech. Rep., Addison-Wesley Inc., 1989.

[27] M. N. Islam, M. Seera, and C. K. Loo, "A robust incremental clustering-based facial feature tracking," *Applied Soft Computing*, vol. 53, pp. 34–44, 2017.

[28] A. Shabtai and Y. Elovici, *A Survey of Data Leakage Detection and Prevention Solutions*, Springer, Berlin, Germany, 2012.

[29] S. R. Kalidindi, S. R. Niezgoda, G. Landi, S. Vachhani, and T. Fast, "A novel framework for building materials knowledge systems," *Computers, Materials and Continua*, vol. 17, no. 2, pp. 103–125, 2010.

[30] A. J. Wang, "Information security models and metrics," in *Proceedings of the 43rd annual southeast regional conference on ACMSE43*, pp. 178–184, 2005.

[31] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[32] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering - A filter solution," in *Proceedings of the 2nd IEEE International Conference on Data Mining, ICDM '02*, pp. 115–122, December 2002.

[33] C.-S. Liu, "An analytical method for computing the one-dimensional backward wave problem," *Computers, Materials and Continua*, vol. 13, no. 3, pp. 219–234, 2010.

[34] G. Katz, Y. Elovici, and B. Shapira, "Coban a context based model for data leakage prevention," *Information Sciences*, vol. 262, pp. 137–158, 2014.

[35] X. Huang, Y. Lu, D. Li, and M. Ma, "A novel mechanism for fast detection of transformed data leakage," *IEEE Access*, vol. 1, pp. 1–11, 2018.

[36] M. Porter, "The porter stemming algorithm," 2006.

[37] F. Pacheco, M. Cerrada, R.-V. Sánchez, D. Cabrera, C. Li, and J. Valente de Oliveira, "Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery," *Expert Systems with Applications*, vol. 71, pp. 69–86, 2017.

[38] "Information week global security survey," Information Week, 2004.

[39] C. M. Praba, "A technical review on data leakage detection and prevention approaches," *Journal of Network Communications and Emerging Technologies (JNCET)*, 2017.

[40] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M. A. Babar, and A. Rashid, "Data exfiltration: A review of external attack vectors and countermeasures," *Journal of Network and Computer Applications*, 2017.

[41] K. Thomas, F. Li, A. Zand et al., "Data Breaches, phishing, or malware? understanding the risks of stolen credentials," in *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security, CCS '17*, pp. 1421–1434, November 2017.

[42] J. DeBlasio, S. Savage, G. M. Voelker, and A. C. Snoeren, "Tripwire: Inferring internet site compromise," in *Proceedings of the IMC '17*, pp. 1–14, 2017.

[43] W. Xu, S. Xiang, and V. Sachnev, "A cryptograph domain image retrieval method based on paillier homomorphic block encryption," *Computers Materials and Continua*, pp. 1–11, 2018.

[44] J. Cui, Y. Zhang, Z. Cai, A. Liu, and Y. Li, "Securing display path for security-sensitive applications on mobile devices," *Computers, Materials and Continua*, vol. 55, no. 1, pp. 17–35, 2018.

[45] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless

sensor networks," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2018.

[46] D. Wang, H. Cheng, P. Wang, J. Yan, and X. Huang, "A security analysis of honeywords," in *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium*, pp. 18–21, 2018.