ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

A Novel Mechanism for Progressive Data Leakage Detection

K.Kathiresan[1], K.Aarthy[2], K.Agalya[3], R.Gowsalya[4], S.Nisha[5]

AP, Department of CSE, Angel college of Engineering and Technology, Tamil Nadu, India [1]

IV year CSE, Angel college of Engineering and Technology, Tamil Nadu, India [2,3,4,5]

ABSTRACT: A data distributor has given sensitive data to a set of supposedly trusted agents. Some of the data is leaked and found in an unauthorized files. The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. We present a novel, progressive data leakage detection framework that significantly increase the efficiency of finding duplicates. Our near solves this difficulty in three pace. 1) pair-selection of words using Bag-of words, 2) word pair-wise comparison using word jumbles , and 3) random walk between word blocks. The proposed solution aims to detect large amounts of newly generated, extensively transformed data accurately and efficiently.

I. INTRODUCTION

Data are among the most important assets of a company. But due to data changes and sloppy data entry, errors such as duplicate entries might occur, making data cleansing and in particular duplicate detection indispensable. However, the pure size of today's datasets render duplicate detection processes expensive. Online retailers, for example, offer huge catalogs comprising a constantly growing set of items from many different suppliers. Progressive duplicate detection identifies most duplicate pairs early in the detection process. Instead of reducing the overall time needed to finish the entire process, progressive approaches try to reduce the average time after which a duplicate is found. Early termination, in particular, then yields more completes results on a progressive algorithm than on any traditional approach.

With incremental algorithm reports new duplicates at an almost constant frequency. This output behavior is common for state-of-the-art duplicate detection algorithms. In this work, however, we focus on progressive algorithms, which try to report most matches early on, while possibly slightly increasing their overall runtime. To achieve this, they need to estimate the similarity of all comparison candidates in order to compare most promising record pairs first.

With the pair selection techniques of the duplicate detection process, there exists a trade-off between the amount of time needed to run a duplicate detection algorithm and the completeness of the results. Progressive techniques make this trade-off more beneficial as they deliver more complete results in shorter amounts of time. Furthermore, they make it easier for the user to define this trade-off, because the detection time or result size can directly be specified instead of parameters whose influence on detection time and result size is hard to guess.

II. RELATED WORK

In this segment, connected study work about data leakage Detection is abridged. Existing study can be divided into two division: adaptive weighted graph learning and score walk.

A. ADAPTIVE WEIGHTED GRAPHS LEARNING

The word vector space model represents text by vectors, which is widely used in many text-related tasks, such as information retrieval and sentiment analysis. The graphs are another method which also has been used for text

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

representation in many text analysis tasks. Different from the tasks of natural language processing (NLP), the sensitivities of terms are more concerned than their meanings in data leak detection. Graphs can better keep the structure information of a document besides its content. We use adaptive weighted graphs to retain the sensitive semantic of documents.

We define the adaptive weighted graph based on
Traditional graphs in Definition 1.
Definition *1 (Adaptive Weighted Graphs):* Let *V* be the set of nodes (key terms), and each node has a value of text term and a weight *wn*. The set of node weights is *WN* .
.
An edge *e* connecting two nodes denotes that they have a certain degree of correlation, and each edge has a weight *we* to quantify this correlation degree. Let *WE* be the set of edge weights. Weight matrix *A* consists of elements *w* 2 *WN* [*WE*, which contains the entire structure and weight information
of the graph. An adaptive weighted graph is denoted as *G* D f*V*; *A*g.

An adaptive weighted graph is a summary of a document which keeps the key terms and their contextual information of the document. In the learning phase, how to retain as much key information as possible and simplify the graph to minimize the amount of subsequent calculation
is the main concern. The adaptive weighted graph provides a balance between the two points mentioned above.

Moreover, what DLD concerns with is not the word semantic, but the sensitivity of the content. We call it sensitivity semantic to distinguish from the words semantic used in NLP. The sensitivity semantic defined in AGW focuses more on the appearance of key terms and their contextual correlations.

B.SCORE WALK ALGORITHM

A score walk algorithm is proposed in this section to detect the sensitive data. In the detection phase, there are mainly two challenges: 1) how to detect the large amounts of transformed data? 2) how to detect the large amounts of fresh data? We transfer these two problems to a graph matching problem in AGW. The two graphs may constructed from two similar documents, but one of which may be transformed or fresh to the other. How to match such two graphs to further detect sensitive data is the main problem we need to tackle. Since the
goal of detection is to calculate the sensitivity of each document, the graphs should be tested separately instead of being merged.

The documents to be tested are first processed
according to graphs learning Algorithm 2 except step 8, *D*0 ! *G*0, where *G*0 D fG0 *d1 ; : : : ;*G0 dn g is the set of weighted graphs. Then the improved label propagation proposed in Algorithm 1 is applied on each graph *G*0 *d* respectively. Nodes belong to both *G*0 d and template *G* are first initialized by their category label in *G*, while the others are initialized randomly. The rest of improved label propagation is performed as in Algorithm 1. At this stage, all the tested documents have been converted into weighted graphs.

The score walk algorithm is proposed to solve the problem of graph matching between tested graphs *G*0 and template graph *G*. For each graph *G*0 d 2 *G*0 to be tested, its nodes and edges are walked through by breadth first traversal to calculate a matching score. A reward and punishment mechanism is used to calculate the score *fw*() during the process of walking. For a node *ni* 2 *G*0
d , if ni 2 G, a node bonus fw(ni;C) as shown in Eq. (11a) will be added. Then all nodes nj connecting to *ni* through edge *eij* will be traversed. If an edge *eij* 2 *G*, an edge bonus *fw*(*eij*;C) as shown in Eq. (11b)

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

will be added. If *eij* =2 *G* but in *G* there exist one node also connecting to *ni* with the same label as *nj*:*label* in *G*0, a label bonus *fw*(*eij*; *lb*;C) as shown in Eq. (11c) will be added. Otherwise a label penalty *fw*(*eij*; *lb*;□) as shown in Eq. (11d) will be subtracted. If *ni* =2 *G*, a node penalty
fw(*ni*;□) as shown in Eq. (11e) and its edges penalty *fw*(*eij*;□) as shown in Eq. (11f) will be together subtracted, to punish the mismatched node and its correlations. The detailed rewards and penalties mentioned above are listed in Eq. (11).
fw(*ni*;+) D *wni* C (*wni*
wTn
i
)
(11a)
fw(*eij*;C) D *weij* C(
weij
weT
ij
) (11b)
fw(*eij*; *lb*;C) D *weij* C *weij* _ (11c)
fw(*eij*; *lb*;□) D _ _ *weij* _ (11d)
fw(*ni*;□) D _ _ *wni* (11e)
fw(*eij*;□) D _ _ *weij* (11f)
The whole graph *G*0 *d* is traversed by the score walk, during which the scores of nodes and edges are calculated in turn and added up. The total sensitivity score of *G*0 *d* is shown in Equation. 12, where (□1)*k*1 *fw*(*ni*; *k*1) denotes the node reward
(*k*1 D 2) or penalty (*k*1 D 1), (□1)*k*2 *fw*(*eij*; *k*2) denotes the reward(*k*2 D 2) or penalty (*k*2 D 1) of edge connecting *ni* and *nj*.
fw(*G*0
d) D
X
*ni*2*G*0
d ;
*eij*2*G*0
d
(□1)*k*1 *fw*(*ni*; *k*1) C (□1)*k*2 *fw*(*eij*; *k*2) (12)
Algorithm. 3 provides the detail of score walk algorithm. The sensitivity score of a weighted graph is calculated by one time score walk on the graph, which can perform the detection quickly and efficiently. The final sensitivity is determined by comparing the two score toward sensitive template *GS* and non-sensitive template *GN* . What's more, through the mechanism of reward and punishment, it can perform a general match between the tested graphs and template graph, which can tolerate the transformed or fresh nodes and edges in tested graphs. Thus, score walk algorithm can deal with the large amounts of transformed data and fresh data.

III. MODEL

In this segment, the proposed Progressive data leakage detection consists of three phases: bag After transforming the text into a "bag of words", features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text of words, tokenization and random walk. The Bag-of-words model is mainly used as a tool of feature generation. Jumbles of word counts allow us to compare documents and gauge their similarities for applications . Progressive Random Walk builds upon an central blocking technique and the successive enlargement of blocks.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

A.BAG OF WORDS

A problem with modeling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed-length inputs and outputs. Machine learning algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers. In language processing, the vectors are derived from textual data, in order to reflect various linguistic properties of the text. This is called feature extraction or feature encoding. A popular and simple method of feature extraction with text data is called the bag-of-words model of text. The Bag-of-words model is mainly used as a tool of feature generation.. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

1. A vocabulary of known words.

2. A measure of the presence of known words.

Example of the Bag-of-Words Model

Let's make the bag-of-words model concrete with a worked example.

Step 1: Collect Data

It	was	the	best	of	times,
it	was	the	worst	of	times,
it	was	the	age	of	wisdom,

it was the age of foolishness,

For this small example, let's treat each line as a separate "document" and the 4 lines as our entire corpus of documents.

Step 2: Design the Vocabulary

Now we can make a list of all of the words in our model vocabulary.

The unique words here, ignoring case and punctuation are:

- "it"
- "was"
- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

That is a vocabulary of 10 words from a corpus containing 24 words.

Step 3: Create Document Vectors

The next step is to score the words in each document.

The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model.

Because we know the vocabulary has 10 words, we can use a fixed-length document representation of 10, with one position in the vector to score each word.

The simplest scoring method is to mark the presence of words as a Boolean value, 0 for absent, 1 for present.

Using the arbitrary ordering of words listed above in our vocabulary, we can step through the first document ("It was the best of times") and convert it into a binary vector.

The scoring of the document would look as follows:

- "it"
- "was"

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

1 [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
"it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
2 "it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
3 "it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

All ordering of the words is nominally discarded and we have a consistent way of extracting features from any document in our corpus, ready for use in modeling.
New documents that overlap with the vocabulary of known words, but may contain words outside of the vocabulary, can still be encoded, where only the occurrence of known words are scored and unknown words are ignored. After transforming the text into a "bag of words", features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text. The scores for frequent words and frequent across all documents are penalized. The scores are a weighting where not all words are equally as important or interesting. The scores have the effect of highlighting words that are contain useful information in a given document.

B.JUMBLES WORDS

Jumbles of word counts allow us to compare documents and gauge their similarities for applications like search, document classification and topic modeling. Document classification is a example of machine learning in the from of natural language processing. By classify text, we are aiming to assign one are more classes or categories to a document, marking it easier to manage and sort. Topic model is a type of statistical model for discovering the abstract "topics" that occur in the collection of document. Topic modeling is the frequently used text-mining tool for discovery of hidden semantic structures in the text body.

C.RANDOM WALK

Progressive Random Walk (PRW) is a novel approach that builds upon an central blocking technique and the successive enlargement of blocks.

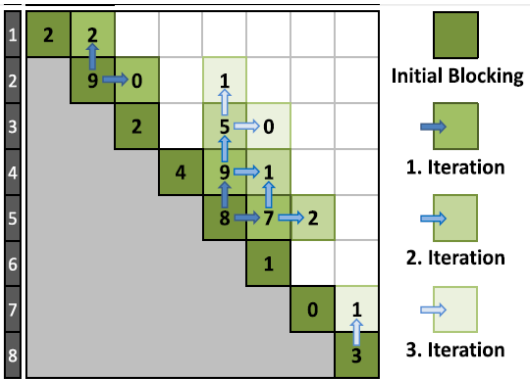
ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019



Above Figure illustrates how PRW chooses matrix. To create this matrix, a pre-processing step has already sorted the records that form the Blocks 1-8 (depicted as vertical and horizontal axes). Each block within the block comparison matrix represents the comparisons of all records in one block with all records in another block. This novel duplicate detection results better data leakage detection than the existing technique.

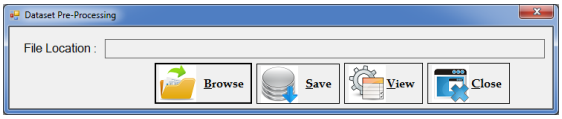
IV. EVALUATION

In this segment, the presentation of progressive data leakage detection is assess by a series of demonstration.

The performance can be evaluated by

- Document preprocessing
- Future extraction
- Implementation of progressive data leakage detection
- Performance comparison

SCREENSHOTS:



1.File Upload

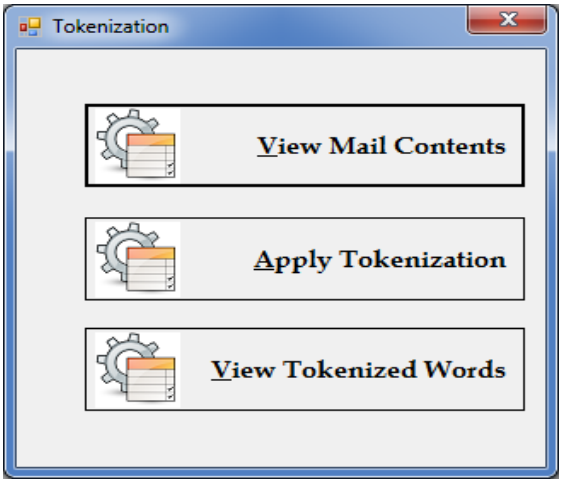
ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

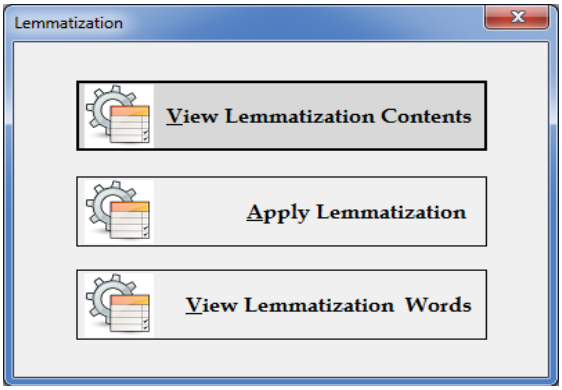
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

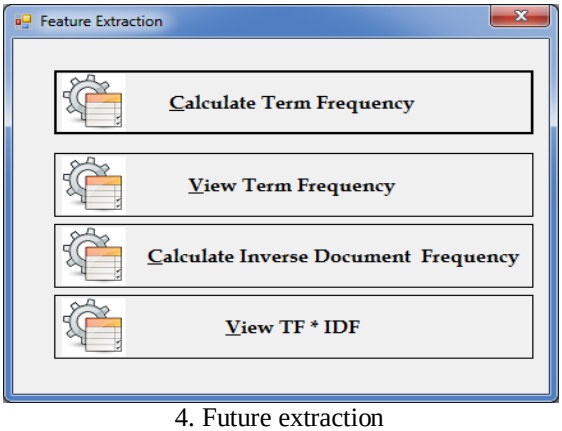
Vol. 8, Issue 3, March 2019



2.Tokenisation



3.Lemmatization



4. Future extraction

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

V. CONCLUSION

In this paper, we present a new method for data leak detection which is progressive data leak detection. This method consist of three phases: pair selection of words using bag of words, word pair wise comparison using word jumbles and random block between word block. The evaluation show that effectiveness of progressive DLD in terms accuracy and scalability. This method can perform the fast detection for transformed or fresh data leaks in real time.

REFERENCES

[1] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, ‘‘A survey on data leakage prevention systems,’’ *J. Netw. Comput. Appl.*, vol. 62, pp. 137–152, Feb. 2016.
[2] Wikileaks. (2010). *Secret US Embassy Cables*. [Online]. Available: https://wikileaks.org/plusd/
[3] B. News. (2013). *Edward Snowden: Leaks That Exposed US spy Programme*. [Online]. Available: http://www.bbc.com/news/world-us-canada23123964
[4] Wikipedia. (2016). *Yahoo! Data Breaches*. [Online]. Available: https://en.wikipedia.org/wiki/Yahoo!_data_breaches
[5] J. Huang, X. Li, C.-C. Xing, W. Wang, K. Hua, and S. Guo, ‘‘DTD: A novel double-track approach to clone detection for RFID-enabled supply chains,’’ *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 134–140, Jan./Mar. 2017.
[6] K. Lab. (2016). *It Security Risks Report 2016*. [Online]. Available: https://www.kaspersky.com/blog/security_risks_report_perception
[7] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, ‘‘Word N-gram based classification for data leakage prevention,’’ in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jul. 2013, pp. 578–585.
[8] S. Xu, ‘‘Bayesian Naïve Bayes classifiers to text classification,’’ *J. Inf. Sci.*, vol. 44, no. 1, pp. 48–59, 2016.