Information Leak Detection in Financial E-mails Using Mail Pattern Analysis under Partial Information

CHETAN KALYAN[1]
Computer Science and Engineering
RNS Institute of Technology
Bangalore-560061
INDIA

KRITHIKA CHANDRASEKARAN
Electronics and Communication Engineering
PES Institution of Technology
Bangalore 560085
INDIA

Abstract:- With the advent of e-mail, sensitive information leakage has become a daunting problem in today's world. Quite often, the mail volume from a company is huge, making manual monitoring impossible. Automatic screening mostly relies on the idea of content scanning, but sometimes the information is so sensitive that even scanning the mails by a third party may not be permitted. Detection under such restrictions becomes difficult. Also, mails originating from specific organizations are often restricted in their subject and content, suggesting that powerful generic techniques like content scanning may not be needed. We propose that selection of proper input variables relevant to the domain could help in such cases; a simple straightforward learning scheme can then detect information leak efficiently using only mail pattern analysis. We used our technique on real life mails from financial institutions. By choosing the input variables judiciously, we were able to learn the mail patterns quite well and detected violations efficiently. The preliminary results are encouraging with an accuracy close to 92%. This technique is now being implemented in a real life commercial tool.

Key-words: Information leak, E-mail violation, Learning algorithms, Neural network, Support vector machine, Pattern recognition

1 Introduction

Information leak via e-mail is a very practical problem. A recent example comes from the Enron incident [1]. In fact, almost all entities dealing in sensitive information, for example financial institutions, military establishments or stock brokers face the formidable challenge of detecting insider information leak. In a typical establishment like this, thousands of e-mails go out to customers and prospective clients every day. The World Talk Corporation estimated in 1999 that over 60 million business people use e-mail [2]; since then the number has only been growing making it practically impossible to manually scan all e-mails for any leak. Automatic detection is the only realistic solution.

Research on information leak violation in e-mails has been little. Carvalho and Cohen's work [3] seems to be the first attempt to specifically tackle this problem. They look at (recipient, message) pairs and define the worst outliers as the leak recipients. To find the outliers, they analyze the text of e-mails and find similarity between the texts; they also look at the mail statistics associated with the recipients. The data is fed to a neural network for learning. They reported a success rate of 82%.

Other researchers have made attempts to tackle the problem of classifying e-mails automatically [4, 5, 6, 7, 8, 9]. The most common technique is to use keyphrase based rules which, depending on the number of times certain keyphrases occur in the mail, assign a violation score to that mail [10]. The keyphrases are often manually defined by the user. The other popular technique is content analysis, in which an automatic system "reads" all e-mails, extracts the commonly occurring words and creates a probabilistic estimate of bad and good words [8]. Spam filters often use this technique [11]. Some of them use this together with different levels of natural language analysis [12]. There exist systems that rely on a social graph analysis of the mails [13, 6]. In this, for every sender and recipient pair mail statistics over the links (going both ways) are extracted and any variation from these statistics is analyzed. Detecting anomalous behaviour of the senders is another technique that is used [13]. Here, every user's mail pattern behaviour is analyzed and significant variations are identified. The patterns

1 Corresponding author. This research has been supported by Saraansh Software Solutions Pvt. Ltd., Bangalore-560079, India.

are learnt by a learning algorithm like neural network or support vector machine [14]. Mostly, all systems use a combination of all these techniques to detect information leakage in e-mails. There also exist some commercial systems; Symantec, Ironmail and Portauthority are among them [15, 16, 17]. A not very rigorous comparison of these three products may be found at http://www.websense.com/Docs/WhitePapers/WP0106-0506_PerceptLabs.pdf. All of these rely heavily on keyphrase detection, content analysis and digital fingerprinting of the mail, and can therefore tend to become quite slow.

1.1 Partial Information

Unfortunately, none of these techniques can work when the information available for analysis is restricted. Quite often, the institutions do not feel comfortable even when an automatic system scans the e-mails and builds a database of information. We faced this problem when trying to build an automatic information leakage detection system for financial institutions. The clients wanted us to detect information leakage *solely* based on mail patterns and without any *sensitive* information extracted from the mails. In fact, they were not willing to give us the mails, but only any "harmless" attributes that we might ask for. Also, for privacy and legal reasons, they could not scan incoming mails (though they had a spam filter in place). Thus, the requirement was that the automatic system should detect information leakage based *only* on mail pattern behaviour of the employees' sent-mails, and that too *only* from "harmless" attributes (thus content of the mails or attachments and recipient addresses were not available). A similar problem comes up when old mail archives are not available. In such cases, only some attributes that were extracted and stored are available for analysis.

This was a major challenge. Obviously, the usual powerful techniques of context analysis and social graph were inapplicable here. Instead, we needed to extract meaningful input variables from the mails that encapsulate the mail patterns and can be used by a learning algorithm. Also, the fact that we were trying to detect violations in a specific kind of e-mail from a specific type of organization (therefore, possibly quite restricted in subject matter variation) suggested that generic techniques like content analysis may not be needed here.

The rest of this paper is organized as follows. In the next section, we describe our approach to the problem. Section 3 describes the data set. Section 4 gives the results and we conclude the paper in section 5.

2 Learning Mail Patterns

As described above, the major hurdle we faced was lack of information. The mail texts were not available to us; only some of its attributes were. Thus, the focus was to detect any significant violation in the mail patterns and relate that to information leak.

Under such conditions a learning approach seems to be the solution. By a learning approach, we mean the following. Suppose we have a set A of attributes extracted from the mails, together with a class T that defines the type of each mail sampled. The type is a binary variable that is one of {violation, non-violation}. Then, we have a set of attributes or variables for the mails together with the mails' "badness" information, which can then be used for a supervised learning scheme like a neural network or a support vector machine. If the mails chosen to create set A is representative enough and if set A well represents the pattern, then the learning algorithm has a good chance of working well.

The main challenge was choosing A. We should not choose too many attributes as that might hamper generalization, and can reduce speed as well. On the other hand, too few attributes might not be enough to capture all kinds of violations efficiently. As the mails were from a specific domain, therefore using some domain knowledge seemed to be in order.

2.1 The Attributes

We started working with the following assumptions: in any institution, a clear list of clients and official recipients is available; and official mails go out mostly during office hours on weekdays. We studied a large set of attributes from the mails, and analyzed the statistical patterns of mails with respect to these attributes to select an optimal set. We analyzed some of the factors as follows.

Time: One can expect a slow rise of official e-mail volume in the morning, reaching a peak and then slowly falling off towards evening, though there might be a burst during closing hours trying to finish off the day's job, and possibly some late night activity for important tasks. Personal mails will stay interspersed throughout, but there might be slight peaks during early morning, evening and around lunch breaks.

A histogram of the number of mails versus time verifies the assumption. Therefore, time is an important attribute. However, working with the *exact* time did not seem very practical as it becomes too specific. Instead, as the histogram suggests, a less fine division

like {morning, afternoon, evening, night} seemed to capture the trend with less confusion.

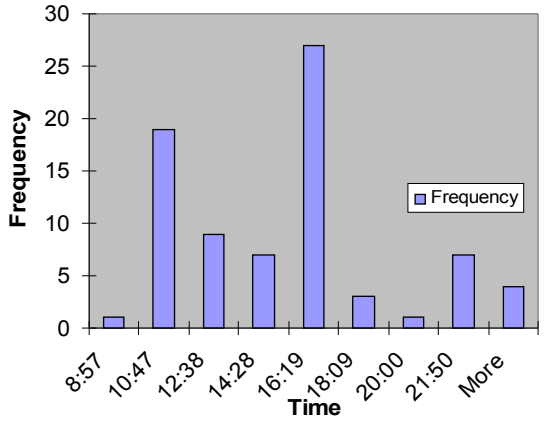

Fig. 1 Histogram of official mails versus time

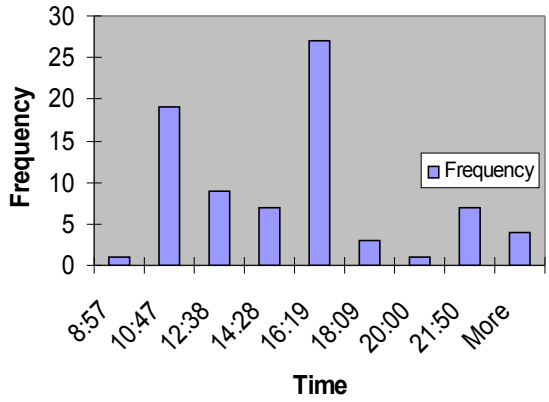

Fig. 2 Histogram of personal mails with time

Attachment: One can also expect the attachment type and size to reveal some information. It is unusual for a personal mail to have a spreadsheet attachment, while an official e-mail is unlikely to have an image attachment unless it is greetings time. Also, it is likely that more violations will occur in mails with attachments (leaking important company information). The following histogram shows a typical result. Note that most official mails have a typical attachment size of about 50 KB, with very few mails having an attachment size more than 150 KB.

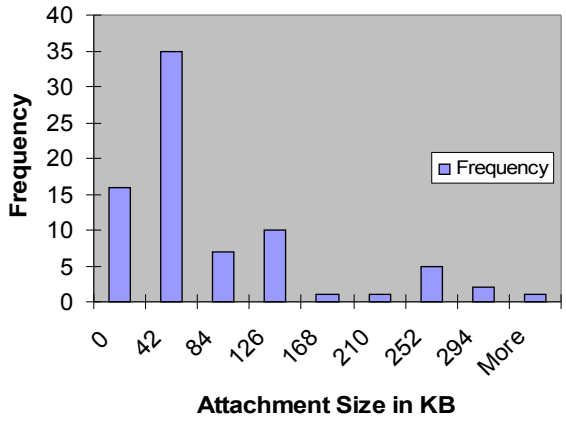

Fig. 3. Histogram of official mails vs. attachment size

Size: The e-mail size is important. We noted that the typical size of an official e-mail (excluding the attachment) rarely exceeds a certain limit. In personal mails too, large "only-text" mails are extremely rare.

Mail Reply Type: Whether the mail is a reply to an older mail, or a forward or a newly composed mail is important information. For example, if the employee is replying to an earlier clean mail with no additional address attached, it is likely to be a good mail. Read receipts and auto-replies are expected to be clean. If a client's mail is being forwarded to another official, that too is possibly clean. On the other hand, if a client's mail is being forwarded to some unexpected personal address, it is probably malicious.

Salutation and Ending: How a sender is addressing the recipient and how the sender is signing off can be a good indicator. Again, choosing the exact salutation or ending is not a good idea; it can vary a lot even between the same (sender, recipient) pair. Instead, we can use the "tone" of the salutation or ending: In official mails, we are likely to find very formal salutations and endings, and mostly informal ones in personal mails. It is likely that some simple forwards or violations may not even carry a salutation or ending. Note that, though this means "reading" the mail, this is but harmless information.

Others: Other important parameters include: if the mail contains a CC, and if so, if the CC is to an official address or a personal address; if it contains a Bcc and so on. For example, a Bcc to any address is usually suspicious, especially if it contains any attachment.

Combining all these, we finally settled on 17 attributes for the supervised learning scheme. We toyed with several combinations, and after some trial and error, this set performed best.

Attribute	Meaning	Possible values
Personal/official id	If the recipient's address is Personal/Official	{personal, official}
Time_slot	Time of day when mail was sent	{morning, afternoon, evening, night}
Day_class	Day when mail was sent	{weekday, weekend}
Subject_class	The category of subject	{new, forward, reply, autoreply, none}
Contains_forward	If the mail contained the forwarded mail	{yes, no, irrelevant}
Contains_new	If the mail contained new words written by sender	{yes, no, irrelevant}
Salutation_class	What kind of salutation the sender used	{formal, informal, none}
Ending_class	What kind of ending the sender used	{formal, informal, none}
CC	If the mail had CC	{yes, no}
CC contains	What kind of addresses the CC contained	{personal, official, none, mixed}
CC to	How many recipients were there in the CC list	{single, multiple, none}
Bcc	If the mail had BCC	{yes, no}
Mail body size	Size of mail body	Continuous
Attachment	If the mail had attachment	{yes, no}
Attachment size	Size of attachment	Continuous
Attachment type	The file type of attachment	The file type
Personal/Official	If the mail is personal/official	{personal, official}

Here some explanation is needed for the *personal/official* variable. This seems to be a very content specific information; how can one decide if a mail is personal or official unless one "reads" that mail? This is true in the strict sense, but for our purpose, this information was extracted by looking at the attachment type, attachment name (official file names are available in a list), the recipient/CC addresses etc. However, to avoid any possible pitfall, we analyzed our data both with and without this variable; these will be presented in results section.

Note that most of the attributes are categorical. This is necessary because we needed attributes that will capture the behaviour pattern of the sender but will not be too specific. For example, we could not take the exact time of sending of the mail; that will be too fine and will confuse the training of the learning algorithm. Also, we could not use exact words found in the mail (for example, the salutation) as those can vary from person to person, and even for the same person when writing to different recipients. The subject line was taboo (it may contain critical information), so we needed to define an attribute that would encapsulate the sender's intent. We achieved this by classifying each mail as one of {forward, reply, new, autoreply}.

A stepwise discriminant analysis was performed on this set to find the most important contributors. A stepwise discriminant analysis needed continuous variables and thus every categorical attribute was converted to several continuous ones. This process does select a small subset of these variables (9). However, we decided to use all 17, as domain knowledge of the problem seemed to point in that direction. For example, the discriminant analysis used only the html type attachment whereas we felt spreadsheet attachments were more important.

Most of these attributes can be extracted by an automatic script without violating the restriction on scanning. The personal/official-id is obtained from a list maintained in the institution (the clients and prospects are all listed). Only the final attribute (if the mail is personal/official) needed manual scan which the institution provided together with the {violation, non-violation} information.

We extracted these attributes and used them for two different learning algorithms: a neural network (multilayer perceptron) with one hidden layer with 10 neurons; and a support vector machine. These two were chosen as they seemed to give the best results. The discrete attributes were converted to continuous ones, which made a total of 53 attributes.

3 The Data set

We had 554 mails at our disposal; totalling 82.5 MB. Naturally, most of these mails were clean (i.e. without

violations). We had 70 violation mails in this set of 554. What is the right number of training examples for the network? The following empirical result was of help [18].

Proposition 1 (Lange et al, 1995) *For a neural network with W weights, the minimum training set size is between [5W/16, 6W/16], with perfect generalization above 6W/16.*

The neural network had 54 continuous inputs (including a constant bias), and 10 hidden neurons. Thus the number of weights is 562. Then by the above proposition, the minimum training size is about 175. We extracted attributes from all 554 mails and then used two methods, namely a neural network and a support vector machine (linear kernel) to learn the patterns. Assessment was done by cross-validation, bootstrapping and random train-test (with about 2/3 rd of these 554 mails for training, thus getting 369 samples for training, which is good by the above proposition).

4 Results

We used Tanagra, a free software for data mining in our experiments [19]. Part of the results was also cross-checked using Orange [20] and NNinExcel [21]. The following table shows the results. The figures in parentheses show the results obtained *without* the personal/official variable.

Algorithm	Measure	Error	Confusion Matrix		
Multilayer Perceptron, with 554 mails	Overall	1.62% (1.81 %)		No	Yes
			No	482 (481)	2 (3)
			Yes	7 (7)	63 (63)
	Cross Validation, 2 fold, 5 repeats	7.08% (8.30 %)		No	Yes
			No	2297 (2293)	123 (127)
			Yes	73 (103)	277 (247)
	Bootstrap, 25 repeats	4.98% (6.75 %)	_____		
	Train-Test, 70% train, 5 repeats	7.31% (8.26 %)		No	Yes
			No	679 (682)	42 (40)
			Yes	19 (29)	95 (84)
Support Vector Machine, with 554 mails	Overall	3.43% (4.87 %)		No	Yes
			No	479 (474)	5 (10)
			Yes	14 (17)	56 (53)
	Cross Validation, 2 fold, 5 repeats	5.27% (7.80 %)		No	Yes
			No	2362 (2333)	58 (67)
			Yes	88 (129)	262 (221)
	Bootstrap, 25 repeats	4.88% (6.55 %)	_____		
	Train-Test, 70% train, 5 repeats	5.51% (6.23 %)		No	Yes
			No	703 (709)	15 (18)
			Yes	31 (34)	86 (74)

As expected, the detection rate suffers without the personal/official variable, but even then the accuracy is close to 92%.

The error during training and validation while using multilayer perceptron are shown below. For both training and validation, the error falls quite rapidly. Validation error shows some movement but tends to stabilize after a few epochs.

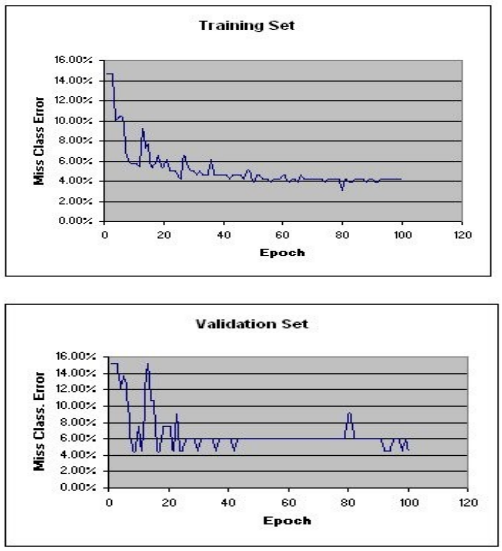

Fig. 4 Error curve for training and validation with multilayer perceptron, one hidden layer, 10 neurons

5 Conclusion

Detecting information leak in e-mails is a major and extremely important practical problem faced by financial institutions. For privacy and sensitivity issues, financial institutions do not feel comfortable in releasing the *exact* content of the mails. We faced this

problem while building a system for some financial institutions. Under such circumstances, detection needs to be done using selected less critical attributes of the mails that reduce the efficiency of keyphrase or content-based systems.

We show that proper choice of input variables is extremely important in such cases, and if this can be done, then learning algorithms can detect leakage quite efficiently. We used real life data for testing our scheme and we detected almost 92% of the violations correctly with properly chosen attributes, superior to generic content analysis. Selection of these critical attributes takes some effort; statistical analysis together with domain knowledge help choosing the most effective set. This seems to be the first effort along these lines; using public attributes for restricted domain and achieving better accuracy, justifying it with experiments. We believe that this method will be very useful in future research in this area.

Currently, this idea is being implemented by a commercial software. We are experimenting with the current set of attributes in domains other than financial mails. We are also looking for other sets of attributes to achieve better results if possible.

Acknowledgments: We thank Saraansh Software Solutions Pvt. Ltd., Bangalore, India for giving us the opportunity to work on this problem. We also thank Abhi Dattsharma and Kiran for useful discussions.

References:

[1] Boufaden N., Elazmeh, W., Ma Y., Matwin S., El-Kadri, N., and Japkowicz, N., (2005), "Peep—an information extraction base approach for privacy protection in email", Conference on Email and Anti-Spam (CEAS'2005).

[2] Harris, D. and Clark, H. (1999), "Worldtalk releases first Internet e-mail corporate usage report; concludes e-mail abuse at epidemic levels." (http://www.worldtalk.com/Corporate%20Information/press%20releases/iecur.htm)

[3] Carvalho, Victor and Cohen, William, (2007), "Preventing Information Leaks in E-mail", SIAM International Conference on Data Mining 2007, (http://www.cs.cmu.edu/~wcohen/postscript/sdm-2007-leak.pdf)

[4] Helfman, J., and Isbell, C., (1995), Ishmail: Immediate identification of important information. (http://www.research.att.com/~jon/ishmail)

[5] Rennie, Jason, (2000), "ifile: An application of Machine Learning in E-mail Filtering", KDD-2000 Text Mining Workshop, Boston. (http://www.ai.mit.edu/~jrennie/ifile)

[6] Cohen, William, (1996), "Learning rules that classify e-mail", AAAI Spring Symposium on Machine Learning in Information Access.

[7] Scott S.and Matwin. S., (1999), "Feature engineering for text classification", Proceedings of Sixteenth International Conference on Machine Learning, ICML-99.

[8] Segal R. B., and Kephart J. O., (1999), "Mailcat: An intelligent assistant for organizing e-mail", Proceedings of the Third International Conference on Autonomous Agents..

[9] Segal R. B. and Kephart J. O., (2000), "Incremental learning in swiftfile", Proceedings of the Seventeenth International Conference on Machine Learning, ICMl-00.

[10] Surfmail, http://www.surfmail.com

[11] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E., (1998) "A Bayesian approach to filtering junk", Proceedings of Workshop on Learning for e-mail, Text Categorization.

[12] D2K, (2003), "Data to Knowledge Text Mining: E-mail Classification", November 2003, Automated Learning Group, NCSA, University of Illinois.

[13] Stolfo, S. J., Hershkop, S., Hu, C., Li, W., Nimeskern, O., and Wang; K., (2006), "Behavior-based modeling and its application to Email analysis"; ACM Transactions on Internet Technology (TOIT), Volume 6, Number 2, May 2006 (http://portal.acm.org/citation.cfm?id=1149125&dl=&coll=&CFID=15151515&CFTOKEN=6184618)

[14] Drucker, H., Wu, D., and Vapnik., V. N., (1999) "Support vector machines for spam categorization", IEEE Transactions, 10(5).

[15] http://www.symantec.com

[16] http://www.ciphertrust.com/products/ironmail/

[17] Portauthority, http://www.websense.com

[18] Lange, R., and Manner, R., (1995), "Quantifying a Critical Training Set size for Generalization and Overfitting Using Teacher Neural Networks", Technical report, University of Mannaheim.

[19] Rakotomalala, Ricco, (2005), "TANAGRA: a free software for research and academic purposes", Proceedings of EGC'2005, RNTI-E-3, vol. 2, pp.697-702, 2005. (In French), (http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html)

[20] Demsar, J, Zupan, B, and Leban, G (2004) , "Orange: From Experimental Machine Learning to Interactive Data Mining", White Paper (www.ailab.si/orange), Faculty of Computer and Information Science, University of Ljubljana.

[21] Neural Network in Excel, (http://www.geocities.com/adotsaha/NNinExcel.html)