A Heuristic Network-Based Approach to Insider Threat Detection

Background

An insider threat is an individual or group with access to an organization's essential network, systems, or data, that deliberately abuses their access for malicious purposes, such as IT sabotage, theft of intellectual property, or fraud. Recent cases of insider threat include individuals such as Robert Hanssen, Bradley Manning, and Edward Snowden. According to the 2014 US State of Cybercrime survey, up to 46% of electronic crimes were committed by insiders. This report also states that 37% of cybercrime cases could not be referred for legal action because the organization could not identify the individual or individuals responsible for the cybercrime.

Research Objective

The aims of this study were two-fold. The first aim was to determine the utility of five different communication-based feature sets to detect and mitigate insiders within an organization. The second aim of this study was to create a user-friendly interface that can analyze content metrics of an organization's communication data to identify insider threats.

Hypothesis

My hypothesis was that insiders have unique communication characteristics that can be used to identify them through communication network analysis.

Methodology

1) Obtain Enron Corpus

2) Corpus to Network Transformation

3) Feature Sets Extraction

4) Machine Learning

5) Analysis of Results and Rule Identification

6) Software Interface Design

1) Obtain Enron Corpus

Enron was an energy company in the 1990s that went bankrupt after some high-ranking individuals within the company were found guilty of insider trading. Their internal communications (along with the communications of other employees within the company) in the months prior to and following Enron's downfall were released and formed the Enron corpus.

I used a specific Enron corpus produced by Shetty and Adibi (2004). This corpus contains 252,759 emails among 151 former employees of Enron, of whom 29 were insiders. I used email data ranging from the month of November, 2000 to October, 2001 (the month in which Enron's insiders were exposed). Table 1 provides an overview of the Enron email communications data.

Table 1

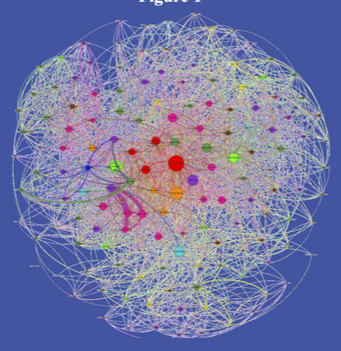
Month	Individuals	Links
Nov-2000	10771	35210
Dec-2000	10905	36535
Jan-2001	6278	22504
Feb-2001	7310	22349
Mar-2001	8624	31929
Apr-2001	11177	35680
May-2001	13989	50290
Jun-2001	10577	36465
Jul-2001	7270	21692
Aug-2001	7702	23383
Sep-2001	10842	31678
Oct-2001	15411	57601

Enron email dataset composition per month (November 2000 to October 2001)

2) Corpus to Network Transformation

I used ORA, a dynamic network assessment and analysis tool, to transform the original corpus to a dynamic meta network, a series of nodes (individuals) with connecting links (emails) that changes over time. I used both "To" and "CC" data. Figure 1 shows a sample Enron network visualization.

Figure 1



Sample Enron network visualization

3) Feature Set Extraction

Five feature sets were used to characterize individual behavior within each dynamic meta-network. For each feature set, pertaining to each variable, I calculated several statistical measures. I used the following feature sets (specific measures within each feature set can be found in the data book):

Feature Set 1: Network Metrics - Social network features that can be used to characterize individual behavior. Calculated in ORA.

Feature Set 2: Network Metric Deltas - The change in network metrics per person from month-to-month to represent behavioral change. Calculated through a coded Java tool.

Feature Set 3: Group-Level Communication Features - Count of the number of interactions from an individual to an entire group (Insiders, non-insider employees, outsiders). Calculated in ORA.

Feature Set 4: Group-Level Communication Deltas - Behavioral change within the group-level communication feature set. Calculated through a coded Java tool.

Feature Set 5: Content Metrics - Analysis of content of the the emails sent within the corpus. Calculated through a coded Java tool.

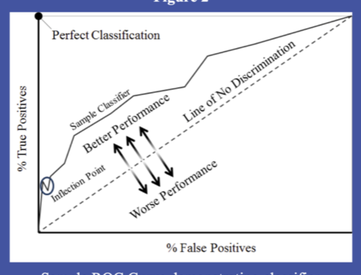
4) Machine Learning

I used Weka to perform supervised machine learning through a rule-based heuristic classifier, JRip, for each feature set. JRip was useful as it provided intelligible results and is robust against over-tuning. This was implemented in a cost-sensitive meta-classifier through five fold cross validation (to allow for at least 5 insiders per test fold). Through adjusting the cost matrix, I was able to produce a series of classification tables that I would later use to create ROC Curves for each feature set.

5) Analysis of Results and Rule Identification

I used Receiver Operator Characteristic (ROC) curves for each feature set to determine the utility of each. This was useful to determine the tradeoff between 1 - specificity (% false positives, number of false classifications of non-insiders as insiders) and sensitivity (% true positives, number of correct classifications of insiders as insiders). Figure 2 shows a sample ROC Curve. To identify the usefulness of each feature set, I calculated the area under the ROC (AUROC), and calculated optimal cutoff points using both unweighted and weighted distances from the line of no-discrimination. A useful feature set would produce an AUROC closer to 1, and would have certain cutoff points with a high true positive rate but low false positive rate. I can then use JRip rules for the optimal cutoff point in my software interface.

Figure 2



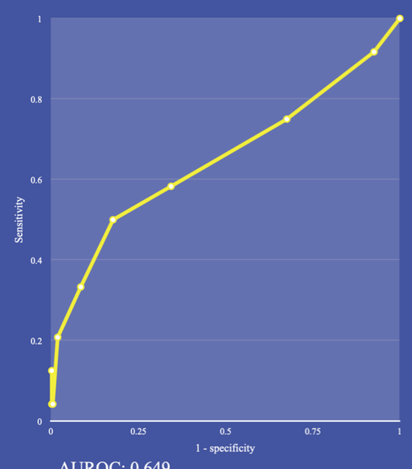
Sample ROC Curve demonstrating classifier effectiveness

6) Software Interface Design

I programmed my software interface in Java, and allowed for customization of rules to suit the user's desired classification accuracy. The program provides not only feedback on individual email content, but also a predictive tool to alert employers of possible insider threats.

Results

Network Metrics ROC Curve

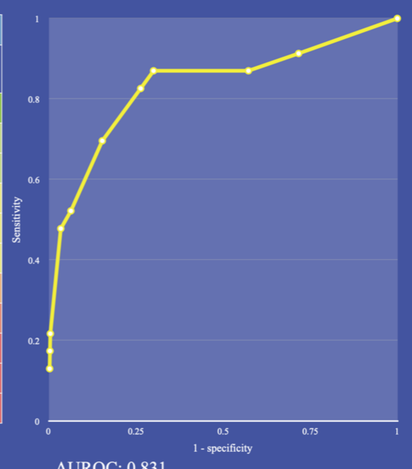


Network Metrics				
Cost	FP	TP	Unweighted Distance	Weighted Distance
1	0.002	0.125	0.123	61.500
10	0.002	0.042	0.040	20.000
100	0.002	0.042	0.040	20.000
1000	0.005	0.042	0.037	7.400
10000	0.019	0.208	0.189	9.947
50000	0.085	0.333	0.248	2.918
100000	0.178	0.500	0.322	1.814
300000	0.344	0.583	0.239	0.695
500000	0.676	0.750	0.074	0.109
1000000	0.926	0.917	-0.009	-0.010
10000000	1.000	1.000	0.000	0.000

AUROC: 0.649

Analysis of network metrics gave an AUROC of 0.649, which is a sub-optimal classification accuracy. The optimal cutoff points were at cost 100000 (unweighted distance) and 1 (weighed distance).

Network Metrics Deltas ROC Curve

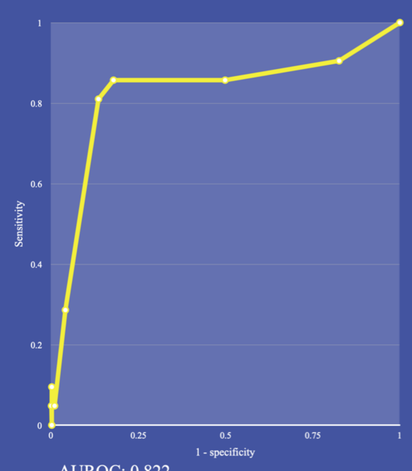


Network Metric Δ				
Cost	FP	TP	Unweighted Distance	Weighted Distance
1	0.003	0.130	0.127	42.333
10	0.004	0.174	0.170	42.500
100	0.005	0.217	0.212	42.400
5000	0.035	0.478	0.443	12.657
10000	0.064	0.522	0.458	7.156
50000	0.154	0.696	0.542	3.519
100000	0.264	0.826	0.562	2.129
150000	0.301	0.870	0.569	1.890
350000	0.573	0.870	0.297	0.518
370000	0.717	0.913	0.196	0.273
400000	1.000	1.000	0.000	0.000

AUROC: 0.831

Analysis of network metric deltas gave an AUROC of 0.831, which is an improvement over the normal network metrics. The optimal cutoff points were at costs 100000, and 150000 (unweighted distance) and 1, 10, and 100 (weighted distance).

Group Level Communication ROC Curve

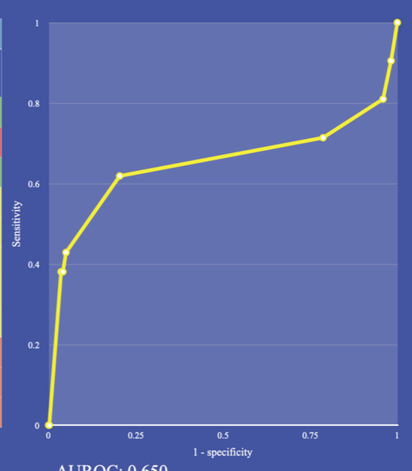


Group Level Communication				
Cost	FP	TP	Unweighted Distance	Weighted Distance
1	0.001	0.048	0.047	47.000
10	0.002	0.000	-0.002	-1.000
100	0.002	0.095	0.093	46.500
1000	0.010	0.048	0.038	3.800
10000	0.041	0.286	0.245	5.976
50000	0.136	0.810	0.674	4.956
100000	0.179	0.857	0.678	3.788
1000000	0.179	0.857	0.678	3.788
1900000	0.499	0.857	0.358	0.717
1990000	0.826	0.905	0.079	0.096
10000000	1.000	1.000	0.000	0.000

AUROC: 0.822

Analysis of group level communication features gave an AUROC of 0.822, which is a good classification accuracy. The optimal cutoff points were at costs 50000, 100000, and 1000000 (unweighted distance) and 1 and 100 (weighted distance).

Group Level Communication Deltas ROC Curve

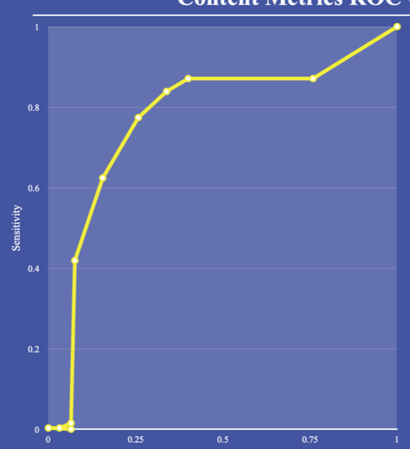


Group Level Communication Δ				
Cost	FP	TP	Unweighted Distance	Weighted Distance
1	0.000	0.000	0.000	
10	0.002	0.000	-0.002	-1.000
100	0.036	0.381	0.345	9.583
200	0.041	0.381	0.340	8.293
250	0.050	0.429	0.379	7.580
500	0.204	0.619	0.415	2.034
550	0.787	0.714	-0.073	-0.093
750	0.959	0.810	-0.149	-0.155
1000	0.982	0.905	-0.077	-0.078
1000000	1.000	1.000	0.000	0.000

AUROC: 0.650

The group level communication deltas gave an AUROC of 0.650, which is worse than the normal group level feature set. The optimal cutoff points were at costs 250 and 500 (unweighted distance), and 100 and 200 (weighted distance).

Content Metrics ROC Curve



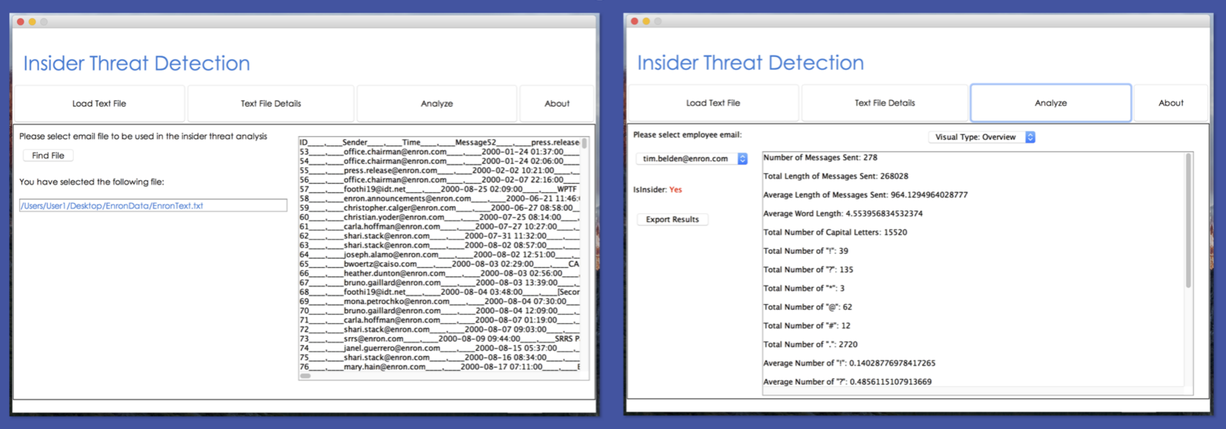
Content Metrics				
Cost	FP	TP	Unweighted Distance	Weighted Distance
1	0.065	0.000	-0.065	-1.000
10	0.000	0.003	0.003	
100	0.832	0.003	-0.029	-0.906
1000	0.065	0.015	-0.050	-0.769
10000	0.076	0.419	0.343	4.513
50000	0.156	0.624	0.468	3.000
75000	0.258	0.774	0.516	2.000
100000	0.339	0.839	0.500	1.475
150000	0.339	0.839	0.500	1.475
300000	0.401	0.871	0.470	1.172
500000	0.759	0.871	0.112	0.148
1000000	1.000	1.000	0.000	0.000

AUROC: 0.771

Analysis of network metric feature set gave an AUROC of 0.771, which is a moderate classification accuracy. The optimal cutoff points were at costs 750000, 100000, and 300000 (unweighted distance) and 10000 (weighted distance).

Insider Threat Detection Interface Screenshots

Figure 3



Features

- File directory selection, in either .csv or .txt format
- File preview included for convenience
- User input for cutoff point selection, from which rules can then be determined
- Node selection option in analysis page
- Analysis presented to the user in both text and graphical form, with rule-based insider threat prediction output included
- Analysis export capabilities of analysis results and specific content metrics for further user inspection

Algorithm Descriptions

Original IREP Algorithm

```
procedure IREP(Pos,Neg)
begin
   Ruleset: = Φ
   while Pos ≠ Φ do
      /* grow and prune a new rule */
      split(Pos, Neg) into (GrowPos, GrowNeg)
        and (PrunePos, PruneNeg)
      Rule:=GrowRule(GrowPos, GrowNeg)
      Rule:=PruneRule(Rule,PrunePose,PruneNeg)
      if the error rate of Rule on
        (PrunePos,PruneNeg) exceeds 50% then
          return Ruleset
      else
          add Rule to Ruleset
          remove examples covered by Rule
            from (Pos,Neg)
      end if
   end while
   return Ruleset
end
```

IREP Optimizations (RIPPER)

1. Alternative rule value metric for assessing value of rules in pruning phase:

v^*(Rule, PrunePos, PruneNeg) ≡ $(p\text{-}n)/(p+n)$

2. Optimized stopping heuristic using description length

RIPPER stops adding rules when description length is more than d bits larger than smallest description length found

3. Post-processing for optimization of rule set

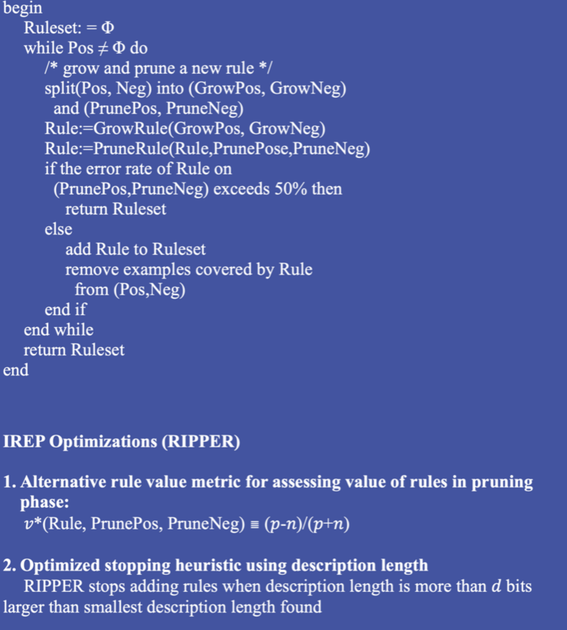
Each rule of a rule set $R_1, \ldots, R_K$ is considered (R_1, R_2, etc.). Two rules are constructed from original rule, replacement rule:

a. Replacement for R_i - Formed by growing and pruning a rule R'_i, while minimizing error of entire rule set ($R_1, \ldots, R'_i, \ldots, R_k$)

b. Revision for R_i - Formed by greedily adding conditions to R_i

Then, the MDL heuristic is used to choose whether to use the original, replacement, or revised rule.

Insider Threat Detection Interface Flowchart



Summary of Results

In summary, my results are as follows:

- Group level communication and network metric delta feature sets provide the optimal classification accuracy within the machine learning classifier
- Content metric feature set provides a moderate classification accuracy
- Delta measures per month may be more useful that raw monthly values for network metrics, but not for group level communication feature sets
- Insider Threat Detection software interface incorporates user-defined cutoff points for rule selection in content metric-based classification

Conclusion

In conclusion, this research presents a novel approach to insider threat detection through the use of communication data, social network analysis, and supervised machine learning. My hypothesis was supported by my results, i.e. the use of several communication-based feature sets, including network metrics, network metric deltas, group level communication, group level communication deltas, and content metrics can be effective in accurately identifying insider threats within an organization. Within these feature-sets, I found that group level communication and network metric delta feature sets have the potential to be the most useful for insider threat detection. Finally, I introduce an interface that can utilize the rules proposed by the machine learning algorithm to enable easy and user-friendly analysis of an organization's communication content data.

Limitations of Study

- Dataset used may not be a complete representation of insider threats, and therefore may limit the findings and results of the research.
- The dataset is relatively old, and therefore may not accurately represent current employee communication patterns.
- Newer means of communication such as text messaging and social media are becoming popular among employees, making email communication less common.

Future Research

- Development of novel feature sets to increase classification accuracy of insider threats, by addressing the limitations addressed above.
- Utilize several datasets for my research to reach a broader range of insider threats in different scenarios, to make my tool more robust and effective.
- Include the analysis of sentiment of email content to perhaps obtain insight on an individual's opinion or attitude to add to the insider threat detection model.

Acknowledgements

I would like to thank the following individuals for their guidance:
Prof. Kathleen M. Carley, CASOS, ISR, SCS, Carnegie Mellon University
Geoffrey Morgan, CASOS, ISR, SCS, Carnegie Mellon University
Mr. Bruce Allen, North Allegheny High School