Temporal Sequence Learning and Data Reduction for Anomaly Detection

By Terran Lane and Carla E. Brodley

Presented by Maja Pusara
ECE Department – Purdue University
February 5, 2004
Intrusion Detection Classification

- **Intrusion Detection Systems**
  - **Host Intrusion Detection Systems - HIDS**
  - **Network Intrusion Detection Systems - NIDS**
Intrusion Detection Classification (Continued)

Intrusion Detection Systems

Signature Matching (Knowledge-Based)

Anomaly Detection (Behavior-Based)
IDS Design Process

Data Collection

Data Normalization

Feature Selection

Classification
Data Sources for HIDS

- Command Line
- System Calls
- Mouse Movement
- GUI Events
- Keystrokes
- ...

CS 590D, 2004
Data Collection

Command line data:

% pwd
% ls
% cat file1.txt file2.txt > file3.txt
% rm file1.txt

Intrusion Detection System
Instance-Based Anomaly Detection System

ls -1af cd foo/ cat bar.c baz.c ...

Tokenize()

ls -1af cd <1> cat <2> ...

Sim0

Profile

Sim0

F0

Profile

Class0

Params

1 1 1 1 0 0 1 0 1 1 ...

CS 590D, 2004
Instance-Based Anomaly Detection System
What is in the Data Set?

Discrete vs. Continuous
Time dependent vs. Time independent
Numerical vs. Non-numerical

% pwd
% ls
% cat <2> > <1>
% rm <1>
What is in the Data Set?

- **Discrete** vs. Continuous
- **Time dependent** vs. Time independent
- Numerical vs. **Non-numerical**

Temporal Sequence Data
Feature Extraction
Similarity Function

• Input:
  – Data stream forming a feature vector of fixed-length L
  – User’s behavioral profile containing D sequences

• Output:
  – Temporal sequence of real-valued similarity measures
Similarity Function (Continued)

Given two sequences of equal length $L$

$$X = (x_0, x_1, \ldots, x_{l-1})$$

$$Y = (y_0, y_1, \ldots, y_{l-1})$$

We compute the following

$$w(X, Y, i) = \begin{cases} 
0 & \text{if } i < 0 \text{ or } x_i \neq y_i \\
1 + w(X, Y, i - 1) & \text{if } x_i = y_i 
\end{cases}$$

$$\text{Sim}(X, Y) = \sum_{i=0}^{l-1} w(X, Y, i).$$
Similarity Function (Continued)

\[
\text{Dist}(X, Y) = \text{Sim}_{\text{max}} - \text{Sim}(X, Y)
\]

\[
\text{Sim}_{\text{max}} = \text{Sim}(X, X) = \sum_{i=1}^{l} i = l(l + 1)/2.
\]

\[
\text{Sim}_D(X) = \max_{Y \in D} \{\text{Sim}(Y, X)\}
\]
Noise Suppression

ls -laf cd foo/ cat bar.c baz.c ...

Tokenize()

ls -laf cd <1> cat <2> ...

Sim0

Profile

I I I 0 1 0 0 1 0 1 1 ...

F0

Class0

Params
Noise Suppression (Continued)

• Input:
  – Results of the similarity function for each sequence over a sliding window of W sequences

• Output:
  – Temporal sequence of smoothed real-valued similarity measures
Noise Suppression (Continued)

- Sliding window of $W$ sequences:

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & \ldots & W & W+1 \\
\end{array}
\]

- 1st $W$ sequences

- 2nd $W$ sequences $\ldots$

- Noise suppression function:

\[
v_D(j) = \frac{1}{W_{i=j-W+1}} \sum_{i=j-W+1}^{j} \text{Sim}_D(i)
\]
Classification
What Learning Method to Apply?

Supervised
- Closed setting
- Labeled data
- Examples:
  - Decision trees
  - Neural networks

Unsupervised
- Open setting
- Non-labeled data
- Examples:
  - Markov models
  - Instance-based learning
IBL Classification

• Input:
  – Temporal sequence of smoothed real-valued similarity measures

• Output:
  – Binary values:
    • 1 is “normal”
    • 0 is “abnormal”
Threshold Selection

![Histogram showing threshold selection](image)

- **U1**: Anomalous User (Unobservable distribution)
- **U0**: Profiled User

Legend:
- Dashed line: Bayes-optimal threshold
- Dotted line: Acceptable False Alarm threshold
Threshold Selection (Continued)

\[ T_{\text{min}} = r/2 \]

\[ T_{\text{max}} = r/2 \]

"Normal" Range

\[ T_{\text{min}} = r/2 \quad r \quad T_{\text{max}} = r/2 \]
Classification Function

• Neyman-Pearson hypothesis test:

\[
\text{class}(v) = \begin{cases} 
1 & \text{if } P_{\{T\}}(v) \geq r \\
0 & \text{if } P_{\{T\}}(v) < r 
\end{cases}
\]

• Acceptance region:
  – Between T-min and T-max
  – Smaller r implies a wider acceptance region
Concept Drift
Empirical Evaluation

• Real-world data sets:
  – 8 UNIX users
  – 7,000 tokens per user
  – Collected over a time period of two years

• Performance criteria:
  – Accuracy
    • Acceptance rate
    • Alarm rate
  – Time to alarm (TTA)
Experimental Setup

• Division of the data set:

<table>
<thead>
<tr>
<th>Complete Training Data: 7000 instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train 1</td>
</tr>
<tr>
<td>Train 2</td>
</tr>
<tr>
<td>Train 3</td>
</tr>
</tbody>
</table>

• Parametric values:
  - Sequence length L=10
  - Window length W=100
  - $r \in \{0.5, 1, 2, 5, 10\}$%
Experimental Results: Accuracy
Experimental Results: TTA
Storage Reduction

- **Instance selection:**
  - Random
  - FIFO
  - LRU
  - LFU

- **Clustering**
  - K-centers
  - Greedy clustering algorithm
Instance Selection Results: Accuracy

- Random
- FIFO
- LRU
- LFU
Instance Selection Results: TTA

Comparative Mean Times to Alarm

- Random
- FIFO
- LRU
- LFU

CS 590D, 2004
Clustering Algorithms

• K-centers algorithm:
  – $K = 2$

Original data set  Clustered data set
Clustering Algorithms (Continued)

- Greedy clustering algorithm:

\[ \text{val}(C) = \sum_{x \in C} \sum_{y \in C} \text{Dist}(x, y) / |C|^2 \]

- Halting criterion for \( \text{val}(C) \):
  - Its 1\textsuperscript{st} derivative must be within \( \epsilon \) of 0 for some \( \epsilon \)

- Similarity between a sequence \( X \) and a cluster is:

\[ \text{Sim}(X, C_{\text{cent}}) \]
Clustering Algorithms (Continued)

- Complete greedy clustering algorithm:
  - Maximizes the analog of mean intracluster distance
    \[
    \text{val}\{C_1, C_2, \ldots, C_n\} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \text{Dist}(C_{i, \text{cent}}, C_{j, \text{cent}})}{n^2}.
    \]

- Halting criterion:
  - Threshold C
  - Small C implies many clusters and vice versa
Experimental Setup

• K-centers:
  – $K = \{50, 75, 100, 125, 150\}$ clusters
  – CPU time constraint = 10,000 cycles

• Greedy clustering algorithm:
  – $C = \{0.25, 0.5, 0.75\}$
  – $\epsilon = 0.005$
Results: Greedy versus K-Centers

K-centers $K = 150, C = 0.25$
Results: Greedy versus Base System

Base system

C = 0.25

Base system
Results: Greedy versus LRU

LRU

C = 0.25, S = 250

LRU
Impersonation Instances

<table>
<thead>
<tr>
<th>USER0</th>
<th>USER1</th>
<th>USER2</th>
<th>USER3</th>
<th>USER4</th>
<th>USER5</th>
<th>USER6</th>
<th>USER7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10 tokens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER0</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER1</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER2</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER3</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td>USER4</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER5</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>USER6</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER7</td>
<td>-</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>100 tokens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER0</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER1</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
</tr>
<tr>
<td>USER2</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>-</td>
</tr>
<tr>
<td>USER3</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>USER4</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER5</td>
<td>I</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>USER6</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER7</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>500 tokens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER0</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td>USER1</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>USER2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td>USER3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>USER4</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>USER5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td>USER6</td>
<td>I</td>
<td>-</td>
<td>I</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td>USER7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

CS 590D, 2004
Summary and Conclusions

• Goal:
  – Identification of potentially malicious intrusions while falsely flagging innocuous actions as rarely as possible

• Approach:
  – Anomaly detection via instance-based learning
  – Storage considerations

• Experimental results:
  – High false alarm rate during classification
  – Greedy clustering algorithm outperformed other storage-reduction algorithms
Future Work

• Online training
• Feature set engineering
  – File names, extensions or types
  – Login session data
• Alternative similarity measures
  – Markov models
• Metalearning and hierarchical classifiers
• Domain knowledge
• Automatic parameter selection