

Semi-Supervised Learning on Email Characteristics for Novel Worm Detection

Steve Martin
steve0@cs.berkeley.edu

Anil Sewani
anil@cs.berkeley.edu

University of California at Berkeley

Learning in Systems

- SLT's can help solve open systems problems.
 - Novelty detection to leverage behavioral invariants.
- Example: examining system logs to detect unsafe states.
 - Can detect failures before they happen in critical services.
- Example (this talk): learning on network behavior to detect novel virus infections.
 - Limit propagation rate for novel viruses to prevent widespread epidemics.
- The technique presented here could be easily applied to other problems.

Improving Classifiers

- Previous work: novelty detection in general is often not enough by itself.
 - False negatives very undesirable.
 - Common solution: make the novelty detector model very sensitive.
 - Tradeoff: Introduces additional false positives.
- Idea: leverage any and all supervisor feedback
 - Use this knowledge to partially label data corpus
 - Filter novelty detection results using a classifier trained on **semi-supervised** data.
- Our results: This works and can decrease false positives by an order of magnitude.

Application: Email Worms

- Email worms have caused billions of dollars of damage.
 - MyDoom, Sobig, LoveLetter, etc all spread by email.
- Signature-based methods are effective against known worms only.
 - However, roughly **25 new Windows viruses a day** were released during 2004!
- Human element slows reaction times.
 - Signature generation can take several hours to several days.
 - Signature acquisition and application by users (or service providers) can take several hours to never.

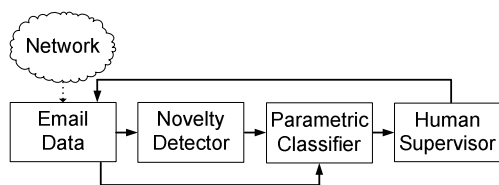
Current Research Solutions

- Need an adaptive first line of defense.
- Employ unsupervised learning on network behavior.
 - Basic idea: leverage behavioral invariant among infected machines.
 - By definition, a worm seeks to propagate itself over a network.
- Novelty detection catches anomalies that could indicate a novel virus.
 - Combine with signature methods.

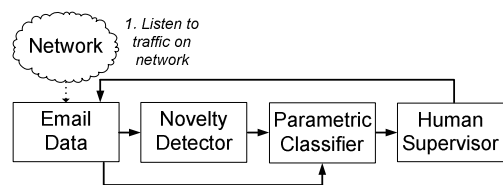
Our Approach

- Novelty detection by itself is not enough.
 - False negatives not acceptable, worm attack will succeed.
 - Too many false positives overwhelm network admins.
- Solution: two-layer approach to filter novelty detector results.
 - Novelty detector minimizes false negatives.
 - Use secondary classifier to filter out false positives.
- Use human reactions to improve secondary classifier.
 - Introduces partial labeling for semi-supervised learning.

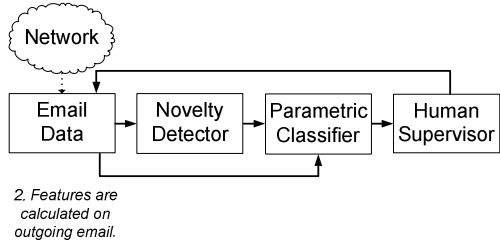
Improved Worm Detection



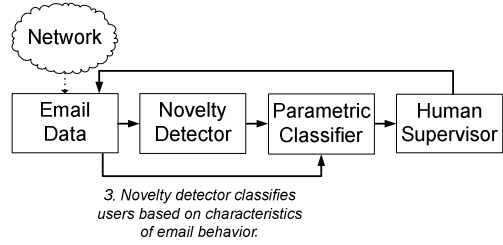
Improved Worm Detection



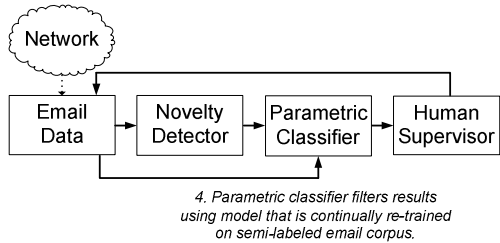
Improved Worm Detection



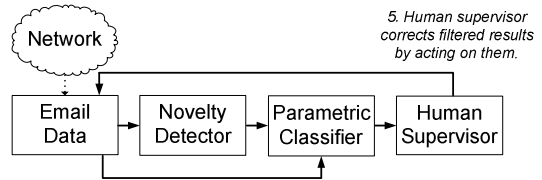
Improved Worm Detection



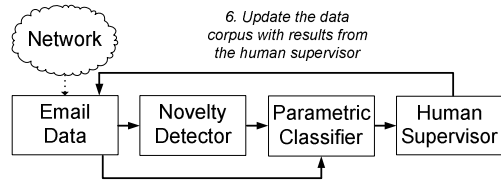
Improved Worm Detection



Improved Worm Detection



Improved Worm Detection



Features

- Distinguished between *per-email* and *per-user* features calculated on outbound email.
- User features capture elements of behavior over a window of time.
 - Examples: mean number of words in emails, ratio of emails sent with/without attachments, etc.
- Email features examine individual snapshots of behavior.
 - Examples: number of characters in the email subject, presence of html in the email, etc.
 - These are then aggregated over a window of time.

Classifier Overview

- Exploit distinct feature distributions using a *generative graphical model*.
 - Fit a specific model for each user.
- Current implementation uses a multi-layer naive Bayes approach.
 - Assumes all features are independent.
 - Allows us to fit specific distributions to infected and non-infected feature data.
- Classifier initially trains on supervised test set.
 - Over time, classifier retrain over *semi-supervised* data.

Feature Distributions

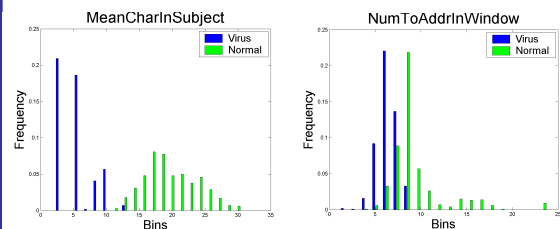


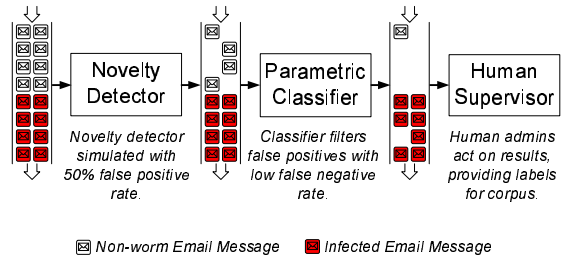
Figure 1. Distributions of the average number of characters in an email subject in Bagle.A

Figure 2. Distributions of the number of address sent to over time in Loveletter.C worm and normal email activity.

Preliminary Evaluation

- Instrumented a mail server to calculate feature data on live email.
 - Conducted a live study with 20 users to collect data and learn distributions of our features.
- Collected virus data from three real worms using mail server and virtual machines.
 - Klez, Loveletter.C, and Bagle.A.
- Constructed training set of real email traffic artificially 'infected' with two viruses.
- Tested on sets of email traffic infected with the third virus as the 'novel' virus.

Current Process



Preliminary Results

Table 1. Parametric Filter Classification Results

Worm Name	Total Emails	# Worm Emails	# Clean Emails	False Positives	False Negatives	Correctly Classified
Bagle	1090	789	301	6 (0.76%)	6 (1.99%)	1078 (98.90%)
Klez	1090	789	301	4 (0.50%)	15 (4.98%)	1071 (98.26%)
LoveLetter	1090	787	303	9 (1.14%)	5 (1.65%)	1076 (98.72%)

- Important observations:
 - Supervisor overhead minimal
 - Time to model convergence is short

Conclusions

- Bottom line: using semi-supervised learning, we can decrease false positives by an order of magnitude.
 - Preliminary evaluation, while synthetic, shows a decrease of 50% false positives to 0.8%.
- Using supervisor feedback could potentially improve results in several settings.
- This work is preliminary; there is much room for improvement!

Some Future Work

- Further analyze individual contributions by features.
 - Perform sensitivity analysis.
- Examine additional features and refine model
 - Incorporate dependencies between features
- Explore possible attacks worms could use to defeat our methods (watch the next talk...)
- Consider other methods of filtering.
 - Our model is parametric; non-parametric methods may work better (decision trees, etc).

Feedback?

Comments and questions welcome!