Masud, M. M. [1], Gao, J.[2], Khan, L.[1], Han, J.[2], Thuraisingham, B[1]
[1]University of Texas at Dallas
[2]University of Illinois at Urbana Champaign

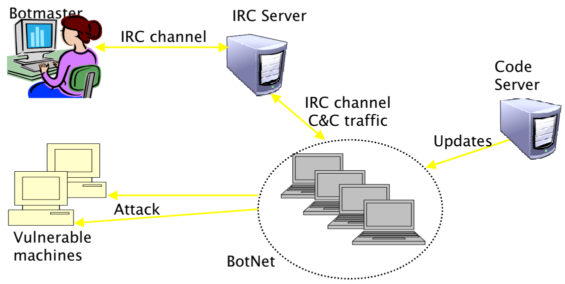
PEER TO PEER BOTNET DETECTION FOR CYBER-SECURITY: A DATA MINING APPROACH

Botnet

Background

- Botnet
  - Network of compromised machines
  - Under the control of a botmaster
- Taxonomy:
  - C&C : Centralized, Distributed etc.
  - Protocol: IRC, HTTP, P2P etc.
  - Rallying mechanism: Hard-coded IP, Dynamic DNS etc.

IRC vs P2P Botnets



- IRC
  - Centralized
  - IRC-based
  - Large
  - Easy to detect
  - CPF – IRC Server
  - Easy to destroy
- P2P
  - Distributed
  - P2P-based
  - Small
  - Hard to detect
  - No CPF
  - Difficult to destroy

Weak Points of P2P Botnets – Rallying Mechanism

- Hard coded IP
  - Trojan.Peacomm (Grizzard et al., 2007)
  - Nugache (Lemos, 2006)
  - Initial Peer list Hard Coded
  - Tries to contact initial peers after infection
  - Can be detected by analysis
- Random IP
  - Sinit (L.T.I. group, 2004)
  - No initial Peer list
  - Probes Random IP
  - Generates a lot of ICMP error

Possible Detection Techniques

- System monitoring
  - Looking for symptoms (e.g. change in “hosts” file)
  - Anti-virus
  - Unusual system calls
- Our approach - Network traffic monitoring
  - Open ports
  - Connection rate
  - Arp requests
  - ICMP errors

What To Monitor?

- Monitor Payload / Header?
- Problems with payload monitoring
  - Privacy
  - Unavailability
  - Encryption/Obfuscation
- Information extracted from Header (features)
  - New connection rate
  - Packet size
  - Upload/Download bandwidth
  - Arp request & ICMP echo reply rate

Mapping to Stream Data Mining

- Stream data : Stream data refers to any continuous flow of data.
  - For example: network traffic / sensor data.
- Properties of stream data : Stream data has two important properties:
  - *infinite length*
  - *concept drift*
- Stream data classification: Because of the two abovementioned properties,
  - stream data classification cannot be done with conventional classification algorithms

Problems with Stream Data Classification

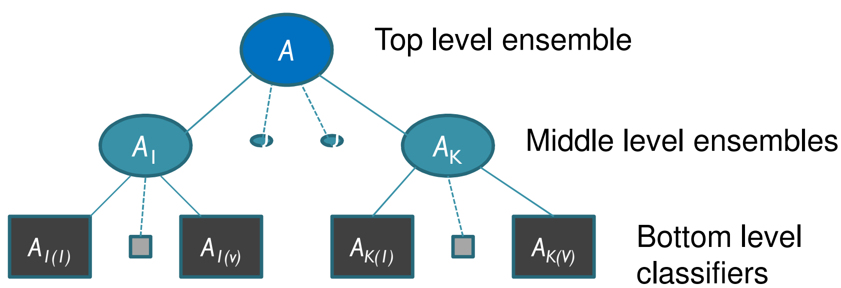
- It is not possible to store infinite amount of historical data for training
- Due to concept drift, characteristics of data changes with time.
  - It is a major problem to choose appropriate training data.
- We propose a multi-chunk multi-level ensemble approach to solve these problems,
  - which significantly reduces error over the single-chunk single-level ensemble approaches (e.g. [1]).

The Single-Chunk Single-Level Ensemble (SCE) Approach

- Divide the data stream into equal sized chunks
  - Train a classifier from each data chunk
  - Keep the best *K* such classifier-ensemble
- Suppose we already have an ensemble of *K* concepts
  - $C = \{c_1, c_2, \ldots, c_K\}$
- Whenever a new data chunk D_i appears,
  - Classify the instances of D_i with *C* using voting
  - Train a classifier c' using D_i
  - $C \leftarrow$ best *K* classifiers among *C* U {*c*' }

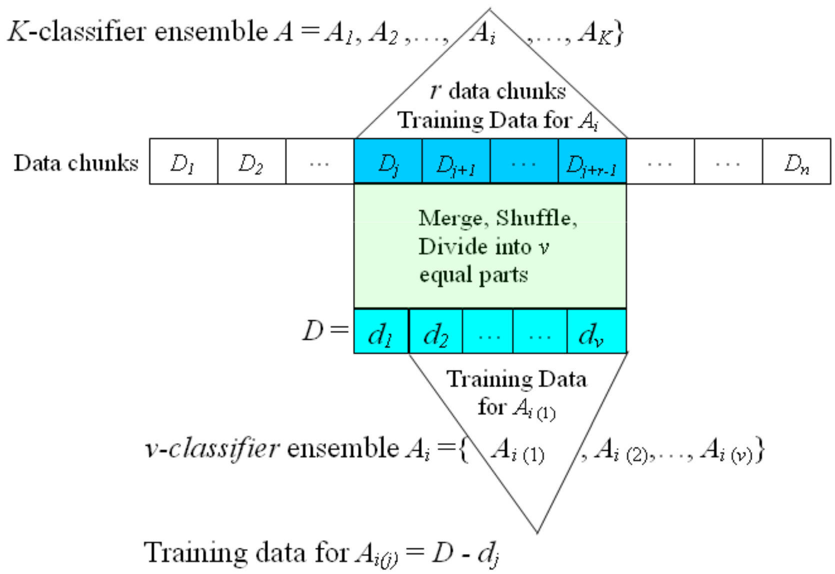
Our Approach: Multi-Chunk Multi-Level Ensemble (MCE)

- Train *v* classifiers from *r* consecutive data chunks, and create an ensemble, and Keep the best *K* such ensembles



- Two-level ensemble hierarchy:
  - Top level (*A*): ensemble of K middle level ensembles *Ai*
  - Middle level (A_i): ensemble of *v* bottom level classifiers $A_{i(j)}$

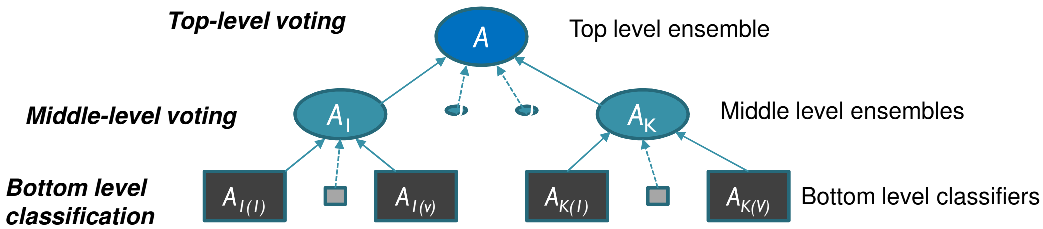
Middle-level Ensemble Construction



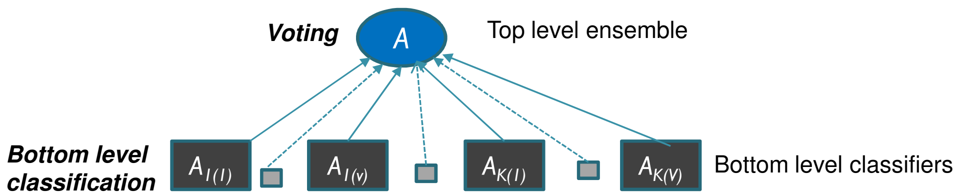
Top Level Ensemble Updating

- Let D_n be the most recent *labeled* data chunk
- Let *A* be the top-level ensemble
- Construct a middle-level ensemble *A`*
  - using *r* consecutive data chunks: $D=\{D_{nr+1},\ldots,D_n\}$
- Obtain error of *A`* on *D* by testing each classifier $A`_{(j)}$ on its corresponding test data d_j
- Obtain error of each middle level ensemble $A_1,\ldots A_k$ on the latest chunk D_n
- $A \leftarrow K$ lowest error middle level ensembles in classifiers in *A* U {*A`*}

Two-Level Voting (MCE) vs Single-Level Voting (MCE2)



Two-Level Voting (MCE)



Single- Level Voting (MCE2)

Error Reduction Analysis

THEOREM 1. *Let σ_C^2 be the error variance of SCE. If there is no concept drift, then the error variance of MCE is at most $1/rv$ times of that of SCE. i.e.,*

$$\sigma_A^2 \le \frac{1}{rv}\sigma_C^2$$

Proof: $\sigma_{A_i}^2 = \frac{1}{v}\sigma_{B_i}^2 \qquad \sigma_{B_i}^2 = \frac{1}{r^2}\sum_{j=i}^{r+i-1}\sigma_{C_j}^2$

$$\sigma_A^2 = \frac{1}{K^2 v}\sum_{i=1}^{K}\frac{1}{r^2}\sum_{j=i}^{r+i-1}\sigma_{C_j}^2$$

$$= \frac{1}{K^2 r^2 v}\sum_{i=1}^{K}\sum_{j=i}^{r+i-1}\sigma_{C_j}^2$$

$$\le \frac{1}{rv}\left(\frac{1}{K^2}\sum_{i=1}^{K}\sigma_{C_i}^2\right), K >= r$$

$$= \frac{1}{rv}\sigma_C^2$$

Error Reduction Analysis (continued)

THEOREM 2. *Let $\hat{\sigma}_A^2$ be the error variance of MCE in the presence of concept drift, σ_C^2 be the error variance of SCE, and P_d be the drifting probability defined above. Then $\hat{\sigma}_A^2$ is bounded by:*

$$\hat{\sigma}_A^2 \leq \frac{(1+P_d)^{r-1}}{rv}\sigma_C^2$$

Proof:

$$\begin{aligned}
\hat{\sigma}_A^2 &= \frac{1}{K^2}\sum_{i=1}^{K}\frac{1}{r^2}\sum_{j=i}^{r+i-1}\hat{\sigma}_{C_j}^2 \\
&= \frac{1}{K^2r^2v}\sum_{i=1}^{K}\sum_{j=i}^{r+i-1}(1+P_d)^{(i+r-1)-j}\sigma_{C_j}^2 \\
&\leq \frac{1}{K^2r^2v}\sum_{i=1}^{K}(1+P_d)^{r-1}\sum_{j=i}^{r+i-1}\sigma_{C_j}^2 \\
&= \frac{(1+P_d)^{r-1}}{K^2r^2v}\sum_{i=1}^{K}\sum_{j=i}^{r+i-1}\sigma_{C_j}^2 \\
&\leq \frac{(1+P_d)^{r-1}}{K^2rv}\sum_{i=1}^{K}\sigma_{C_i}^2, r > 0 \\
&= \frac{(1+P_d)^{r-1}}{rv}\sigma_C^2
\end{aligned}$$

Experiments

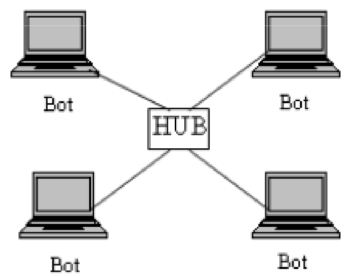
- Synthetic data generation
  - Each data point is a d-dimensional vector $[x_1,\ldots,x_d]$ where $x \in [0,1]$
  - Concept drift is achieved by a moving hyperplane
    - Equation of the hyperplane:

$$\sum_{i=1}^{d} a_i x_i = a_0 \quad , \quad a_0 = \frac{1}{2}\sum_{i=1}^{d} a_i$$

  - Weights are changed at a certain rate
  - Generated 250K data points and created 4 datasets
    - Having 250, 500, 750, and 1000 data points per chunk, respectively

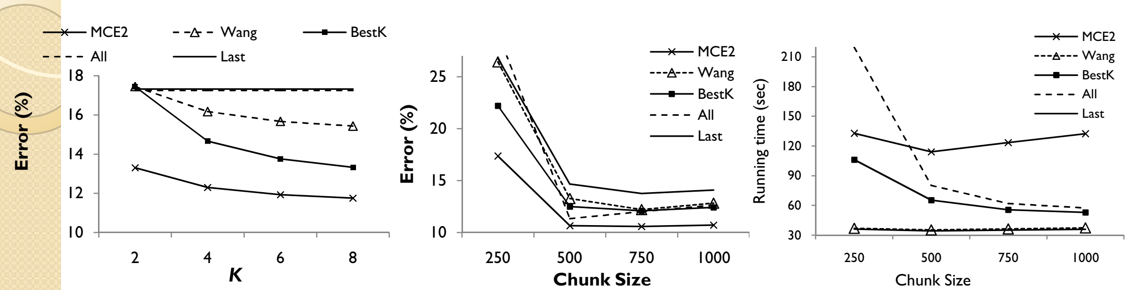
Experiments (continued)

- Botnet data collection
  - Four virtual machines in an isolated environment
    - Running on top of a Windows XP host operating system.
    - Each bot machine is running Nugache (a P2P bot)
  - Collected normal data from uninfected machines

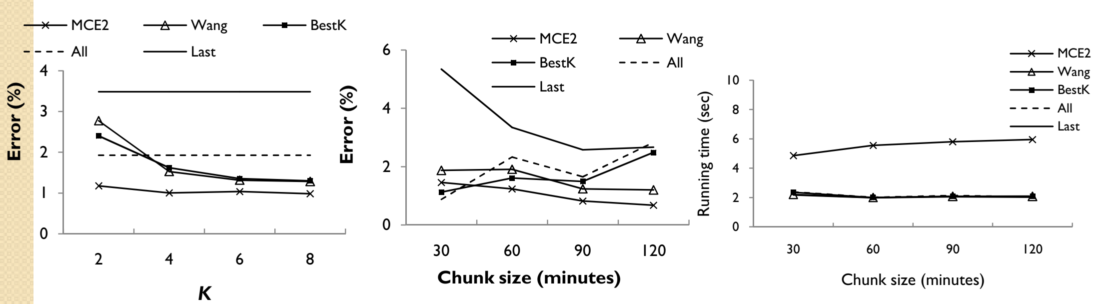


  - Collected 40-hour trace and created 4 datasets
    - Having 30,60,90, and 120-minute data chunks, respectively

Evaluation



Results on synthetic data



Results on botnet data

Evaluation (continued)

Table 1: Error of different approaches on synthetic data using decision tree

Chunk size	M_2	W_2	B_2	M_4	W_4	B_4	M_6	W_6	B_6	M_8	W_8	B_8	*All*	*Last*
250	19.3	26.8	26.9	17.3	26.5	22.1	16.6	26.3	20.4	16.2	26.1	19.5	29.2	26.8
500	11.4	14.8	14.7	10.6	13.2	12.4	10.3	12.7	11.6	10.2	12.4	11.3	11.3	14.7
750	11.1	13.9	13.9	10.6	12.1	11.9	10.3	11.5	11.4	10.3	11.3	11.2	15.8	13.8
1000	11.4	14.3	14.3	10.7	12.8	12.2	10.5	12.2	11.7	10.3	11.9	11.4	12.6	14.1

Table 2: Error of different approaches on synthetic data using Bayes Net

Chunk size	M_2	W_2	B_2	M_4	W_4	B_4	M_6	W_6	B_6	M_8	W_8	B_8	*All*	*Last*
250	20.3	29.3	25.4	18.7	29.0	22.8	18.2	28.9	21.9	17.9	28.8	21.7	32.1	27.1
500	12.7	14.2	14.2	12.4	13.3	13.3	12.3	13.2	13.1	12.1	13.1	12.9	12.9	14.6
750	13.1	14.4	14.4	12.9	13.6	13.5	12.9	13.3	13.3	12.9	13.2	13.3	16.7	15.1
1000	13.0	14.2	14.2	12.7	13.3	13.5	12.6	13.2	13.4	12.5	13.4	13.1	13.6	14.4

Table 3: Error of different approaches on synthetic data using Ripper

Chunk size	M_2	W_2	B_2	M_4	W_4	B_4	M_6	W_6	B_6	M_8	W_8	B_8	*All*	*Last*
250	19.2	26.5	26.0	17.6	26.2	22.4	17.1	26.0	21.3	16.8	25.9	20.9	30.4	26.3
500	11.5	14.2	13.9	10.8	13.0	12.3	10.6	12.6	11.8	10.5	12.5	11.5	11.6	14.1
750	11.0	13.4	13.3	10.6	12.1	12.0	10.5	11.7	11.6	10.5	11.5	11.5	15.7	13.3
1000	11.1	13.8	13.7	10.6	12.5	12.3	10.3	12.1	11.9	10.2	11.9	11.8	12.6	13.6

Conclusion

- We have introduced a multi-chunk multi-level
  - ensemble technique for mining concept-drifting data streams
- We have proven theoretically and empirically
  - that our technique reduced error significantly compared to previous techniques .
- We applied our technique to
  - detect botnet traffic and obtained satisfactory result.
- In future, we would like to apply
  - semi-supervised clustering to label stream data
- References

  - [1] Wang, H., Fan, W., Yu, P., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In *Proc.* KDD, 2003.